برمجة التحكم المنطقى P.L.C.

الجزء الأول

إعداد

ريمون كمال

معهد السالزيان الإيطالى "**دون بوسكو**"

٢شارع عبد القادر طه - الساحل ت: ٢٤٥٧٩٦٥٠ - ٢٤٥٧٦٧٩٤

معهد فنى - معهد صناعى

دورات تدريبية سريعة مركزة

دورات تدريبية تعليمية للمدرسين

المراجع

1. أجهزة التحكم المبرمج وتطبيقاتها العملية.
2. Controllore a logica programmabile P. Bani
3. Siemens Programmable Controller Manual

طبعة جديدة

2011

أسم الكتاب: **برمجة التحكم المنطقى** ***P.L.C.***

الجزء الأول

طباعة:

رقم الإيداع:

الترقيم الدولى:

شكر و إهداء

أهدى هذا الكتاب إلى أبى وأمي الذين لهم كل الفضل بأن أعمل في هذا المجال وهم الذين شجعوني على عمل هذا الكتاب بكل جهد وإخلاص شاكر الله و إياهم وكل من ساهم في تقديمه.

وأشكر أيضاً كل المعلمين الأفاضل الذين ساعدوا على خروج هذا الكتاب إلى المليء.

- ☜ **المدير الإيطالي للمعهد:** الأب رينسو ليوناردوسى
- ☜ **الناظر السابق للمعهد:** الأب بيرناردو أشيربونى
- ☜ **مدير الدورات التدريبية:** أ. ماجد چورچ
- ☜ **أستاذ التحكم:** أ. نبيل رزق – أ. وجية جرجس
- ☜ **أستاذ التكيف والتبريد:** أ. إميل فتح الله
- ☜ **أستاذ الـــ PLC:** أ. ماجد موريس – أ. ماجد عريان – أ. چيوليو جالو – أ. محسن أنتون

أتشرف باستقبال تعليقات السادة القراء بخصوص هذا الكتاب على عنوان البريد الإلكترونى التالى

plcbook@hotmail.com

مقدمة

نظراً للتقدم العلمى السريع المرتبط بالمجال الصناعى وخاصة من الناحية الكهربية أصبح لا غنى عن الربط بين عالم الصناعة وبين التكنولوچيا العصرية ويتمثل هذا الربط بواسطة استخدام أجهزة التحكم المنطقى بمختلف أنوعها والتى تستحق أن تسمى بالأجهزة الذكية نظراً لما تقدمه فى المجال الصناعى من: سهولة فى تصميم البرامج، ومرونة فى أكتشاف الأعطال، ومساعدة فى حل المشاكل، ...الخ

و نظراً لصعوبة ترجمة بعض المصطلحات الخاصة بهذا المجال وخاصة لكى لا تفقد المعنى التقنى أو الفنى لها، تمت كتابتها بلغتها الأصلية لذلك لا تهتم كثيراً عزيزى القارئ بهذه المصطلحات فستكون بسيطة ومفهومة بمجرد ما أن تتعمق بفهم فى هذا المجال.

هكذا أيضاً لا تتعجل عزيزى القارئ فى النظر إلى مواضيع متباعدة خاصة أن كنت بمبتدئ فى هذا المجال وهذا لأنه قد تم شرح المنهج بطريقة متسلسلة ولذلك يفضل للقارئ قراءة المواضيع بالتسلسل التى كتبت به لفهم جميع الأمور دون تخبط.

و خاصاً لفهم التمارين لا يشترط فقط القراءة بترتيب بل يجب أيضاً أن تربط كل شرح و كل رمز بالرسم الموجود ولا تقوم بالقراءة بطريقة عابرة.

تم شرح البرمجة بطريقة عامة دون اللجؤ إلى ماركة بعينها وهذا لكى يخدم كل من يعمل مع وحدات التحكم المنطقى بمختلف أنواعها.

تنقسم معرفة أجهزة التحكم المنطقى إلى أمور عديدة من أهمها:

تصميم برامج – اكتشاف أعطال - حل مشاكل

قد تم التركيز بشكل كبير فى الجزء الأول من هذا الكتاب على معرفة تصميم البرامج بطريقة سلسة وباستخدام أسهل لغات البرمجة.

لذلك أقدم لكم هذا الكتاب لخدمة كل من يدرس أو يعمل فى هذا المجال و أتمنى من الله أن يجد كل من يقرأ هذا الكتاب نفعاً له.

المـــؤلـــف

الباب الأول

جهاز التحكم المنطقى

- ما هو جهـــــــــــــــــــــاز الــ PLC.
- لمـــــــــــــــــــــإذا يستخدم جهاز الــ PLC.
- مكـــــــــــــــــــــونات جهاز الــ PLC.
- تصنيـــــــــــــــــــــف جهاز الــ PLC.
- حمايـــــــــــــــــــــات لجهاز الــ PLC.
- توصـــــــــــــــــــــيل جهاز الــ PLC.
- لمبـــــــــــــــــــــات الإشارة.
- كابل البرمجـــــــــــــــــــــة.
- الذاكرة الداخليـــــــــــــة و الخارجيـــــــــــــة.
- توصيـــــــــــــل وحدة مدخلات و مخرجات أضافية.
- التحـــــــــــــــكم بواسطة كمبيوتر أو شاشة.

ما هو الـــ PLC؟

كلمة PLC هى اختصار لكلمة Programmable Logic Control و هى تعنى برمجة التحكم المنطقى.

صنع أول جهاز تحكم مبرمج فى شركة (جينرال موتورز - general motors) عام 1968. وكان الجهاز فى البداية يحل محل الريليهات التقليدية فقط غير أنه لم يكن قادراً على تحقيق متطلبات الشركة المصنعة ولكنه كان فى الحقيقة بداية لجيل جديد فى صناعة الأجهزة القابلة للبرمجة والتى تطورت فيما بعد، وانتشرت بكثرة فى جميع ميادين الصناعة.

وفى الفترة ما بين عامى 1970 و 1974 ونتيجة للتقدم التكنولوچى فى صناعة الميكروبروسيسور أصبحت الأجهزة القابلة للبرمجة أكثر مرونة و ذكاء، وأصبح من السهل على الفنيين و المهندسين الذين ليس لديهم معرفة كبيرة بعلوم الكمبيوتر و الإلكترونيات الرقمية التعامل معها، بل وأصبحت هذه الأجهزة قادرة على القيام بالعمليات الحسابية و المنطقية و أصبح يمكن التحكم بها باستخدام لغات مختلفة أسهل من التى كانت تستخدم فى ما قبل.

أما فى الفترة ما بين عامى 1975 و 1979 حدث تقدم كبير فى صناعة الأجهزة القابلة للبرمجة، وأشتمل هذا التطور على زيادة سعة الذاكرة وعدد المداخل و المخارج الرقمية بل أشتمل هذا التطور أيضاً على زيادة قدرة الميكروبروسيسور فى سرعة تنفيذ البرنامج.

وكذلك أصبح من السهل تخزين أى برنامج فى وحدة ذاكرة خارجية، وأصبح من الممكن تغيير البيانات سابقة التخزين أثناء التشغيل، فأصبح بوسع وحدة البرمجة تغيير قيم المؤقتات الزمنية المبرمجة والعدادات

المبرمجة و نقلات القيم المتغيرة و مفاتيح المقارنة...الخ، بدون إيقاف خطوط الإنتاج الصناعية كما كان فى السابق.

ملاحظة:

فى بعض مجالات الصناعة لا يمكن لوحدة الـــ PLC التوقف لتعديل البيانات ولذلك فأنه تم التغلب على هذه المشاكل فيما بعد بواسطة وحدات PLC ذات كفاءة أعلى.

ونتيجة لتطور علوم الأتصالات فى هذه الفترة أصبح من الممكن استخدام مجموعة من أجهزة التحكم المنطقى للعمل سوياً فى شبكة محلية للتحكم فى المصنع، كما لو كانت جهازاً واحداً، و أيضاً من الممكن عمل تقارير وافية عن الإنتاج والصيانة و الأعطال بواسطة الوحدات الخارجية التى توصل إلى جهاز الـــ PLC مثل الطابعات أو شاشات التحكم بالمس، وتخدم هذه التقارير إدارات المصانع لتحسين معدل الإنتاج أو تساعد فى الكشف عن الأعطال حيث يمكن طباعة الأعطال التى حدثت فى فترة زمنية معينة.

نظراً للإقبال الشديد فى المجال الصناعى على وحدات البرمجة الذكية PLC تنافست الشركات المصنعة فى تطوير الجهاز وكان نتيجة التطورات الهائلة فى تكنولوچيا صناعة أجهزة التحكم المنطقى ما يلى:

١- أصبحت تكلفة الجهاز منخفضة إلى الحد الذى يسمح باستخدام ه بدلاً من عشرات الريليهات.
٢- أصبح من الممكن استخدام أجهزة التحكم صغيرة الحجم فى التحكم التناظرى analog.
٣- أصبح من الممكن توصيل أجهزة التحكم مع الحساسات الحرارية وأجهزة قياس الانفعال...الخ.
٤- ظهرت أحجام مختلفة من الوحدات المنطقية القابلة للبرمجة فمنها ما يكون عدد مداخله ومخارجه حوالى عشرة فقط، ومنها ما يصل عدد مداخله ومخارجه إلى عدة ألاف.

أدت أيضآ التطورات الهائلة فى أنظمة البرمجة لأجهزة التحكم المبرمج إلى:

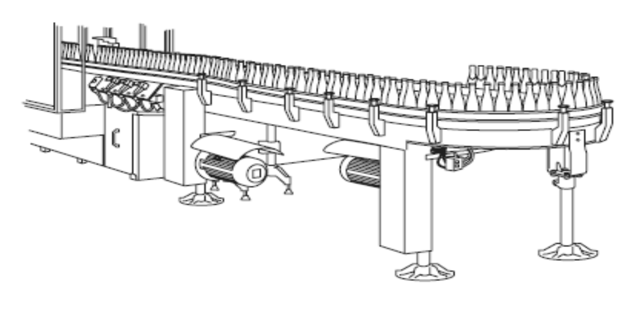
١- استخدام لغات يسهل على من ليس لديه معرفه بعلوم الكمبيوتر استخدامها.

٢- أمكانية تحديد الأعطال وتعديل البيانات الداخلية أثناء تشغيل العملية الصناعية.

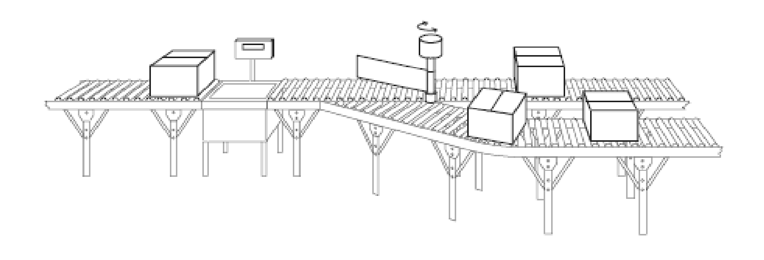
٣- أصبح زمن الاستجابة لأجهزة التحكم المبرمج سريع جداً تصل إلى قراءة البرنامج حوالى 3000 مرة فى الثانية.

٤- أصبحت تستخدم كابلات لنقل المعلومات من و إلى وحدة البرمجة بسرعة فأقة تصل إلى 187,5 Kbps.

٥- أصبح استخدام وحدة الـــ PLC فى المجال الصناعى كثيرآ كما فى الشكل (أ) و الشكل (ب).



الشكل (أ)



الشكل (ب)

مميزات أجهزة التحكم المبرمج:

هناك الكثير من المميزات نذكر منها ما يلى :



١- **التحكم المرن:** والمقصود بالتحكم المرن, سهولة تغيير أداء العملية الصناعية لمواكبة أى توسعات وذلك بتعديل برنامج التشغيل

٢- **الصيانة واكتشاف الأعطال:** إن أجهزة التحكم هى عبارة عن أجهزة ألكترونية لذلك فهى لا تحتاج إلى صيانة وهى معدة لإعطاء بيان عن أعطالها سواء كانت أعطال بسيطة أو أعطال فادحة.

٣- **صغر الحجم مع إمكانيات عالية:** إن أحجام أجهزة التحكم المبرمج صغيرة جداً مقارنة بالدوائر الأخرى فى الكنترول، فيمكن القول أن جهاز تحكم مبرمج أبعاده ١٠سم X ١٥سم X ٢٠سم يمكن أن يحل محل مئات الريلاهات، مئات العدادات، مئات المؤقتات الزمنية, بالإضافة لقدرته العالية للقيام بالعمليات الحسابية بل ويحتوى أيضاً على الكثير من العمليات التى ليس لها مقابل فى الكنترول كما سوف نرى فيما بعد.

٤- **خصائصها لا تتوفر فى أجهزة الكمبيوتر المعتادة:** إن أجهزة التحكم المبرمج معدة للعمل فى البيئة الصناعية التى تتميز بأختلاف كبير فى درجات الحرارة والرطوبة ووجود ضوضاء عالية، وكذلك فهى مصممة على أن يقوم بتركيبها وصيانتها وبرمجتها مهندس الموقع مثل المهندسين الكهربائيين الذين ليس لديهم مهارات خاصة بالالكترونيات الرقمية ولا بعلوم الكمبيوتر.

٥- **يمكن أن تعمل داخل شبكة:** يمكن استخدام مجموعة من وحدات البرمجة المنطقية للتحكم فى الماكينات المختلفة المكونة لخطوط الإنتاج، ثم الربط بين وحدات البرمجة المنطقية بواسطة شبكة محلية يتم من خلالها تبادل البيانات اللازمة للتشغيل، يتم التحكم فى كل منهما باستخدام الكمبيوتر، ويتم تبادل البيانات بين الكمبيوتر و وحدة الـــ PLC من خلال شبكة الاتصالات المصغرة.

وحدة التحكم المبرمج.......PLC unit :

يحتوى جهاز الـــ PLC على وحدة معالجة مركزية CPU وهى التى تقوم بقراءة البرنامج وتنفيذه, حيث تقوم بقراءة الدخل (input) و تطبيق البرنامج ومن ثم تقوم بتشغيل الخرج (output).



لماذا تستخدم وحدة الـــ PLC:

الـــ PLC هو جهاز ذات تيكنولوچيا عالية فلذلك باستخدام الـــ PLC يمكن عمل الكثير من التمارين المعقدة ولكن بطرق بسيطة جداً مقارنة بالكنترول. بل يمكن أيضاً عمل بعض التمارين التى لا يمكن أن تصمم بالكنترول (control) و بخلاف ذالك يوجد مئات المؤقتات الزمنية (timers) و العدادات (counters) و بعض الأوامر الأخرى التى ليس لها مقابل أو مثيل فى الكنترول.

مكونات وحدة الــ PLC:

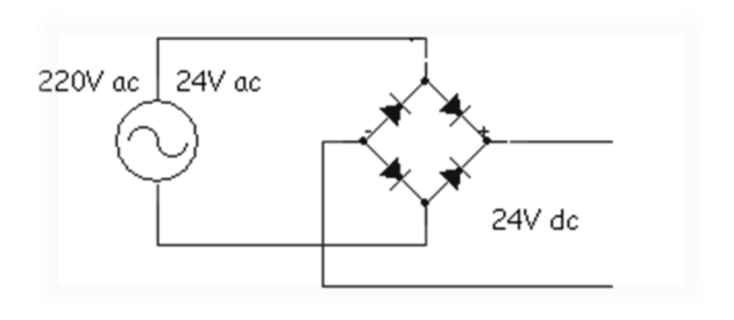
الــ PLC يتكون من:

١- وحدة التغذية..........power supply

وهى مكونه من محول خافض للجهد (step down transformer) و دائرة توحيد (rectifier)

المحول الخافض للجهد يقوم بتحويل جهد التيار المتردد إلى جهد متردد أخر أقل قيمة.

دائرة التوحيد تقوم بتحويل التيار المتردد إلى تيار مستمر دون تغير قيمة الجهد.



ملاحظة:

أ – استخدام وحدة التغذية (power supply) يعتمد على نوع الــ PLC: يوجد بعض الأنواع التى تعمل بــ 220V AC ولذلك فى هذه الحالة يتم التوصيل بمصدر الكهرباء مباشراً ولا يستخدم الــ (power supply).

ب –يحتوى جهاز الــ PLC من الداخل على دائرة توحيد rectifier لتحسين التيار المستمر قدر الأمكان.

ت -في بعض الحالات قد نحتاج أثناء التوصيل إلى استخدام أكثر من وحدة تغذية فى نفس الوقت (power supply) و فى هذه الحالة يفضل أن يتم توصيل الطرف السالب لكل وحدات التغذية المستخدمة معاً لضمان تساوى balance الجهد الخارج من وحدات التغذية.

٢– وحدة المعالجة المركزية..... C.P.U. "Central Processing Unit"

وهى تعتبر العقل المفكر لجهاز الــ PLC وهو الذى يقراء البرنامج ويقوم بالعمليات الحسابية بطريقة فائقة السرعة بحيث يقوم بتشغيل أو فصل الخرج فى الوقت المناسب.

لكل وحدة معالجة مواصفات خاصة تؤثر على سرعتها فى تنفيذ العمليات, فمثلاً:



- المعالج رقم CPU 313: يعمل على قراءة برنامج بحجم 12KB خلال 0.6ms
- المعالج رقم CPU 314: يعمل على قراءة برنامج بحجم 24KB خلال 0.3ms

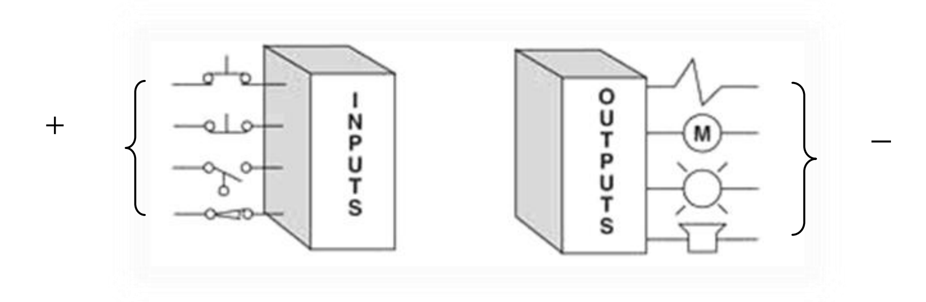
ملاحظة:

- يحتاج المعالج دائماً إلى ذاكرة memory لأحتواء البرنامج بينما يقوم هو بالقراءة والتنفيذ.
- الذاكرة الموجودة داخل الــ PLC دائماً تكون من النوع المتطاير volatile memory وسوف نتكلم عنها بالتفصيل.

٣- وحدة الدخل و الخرج........input and output modules.

بالنسبة إلى الدخل فهو المكان الذى يتم فيه توصيل طرف واحد فقط من طرفى المفاتيح بحيث يقوم بإرسال الإشارة إلى الــ PLC وهذا يطبق على المفاتيح ذات الإشارة الرقمية بجميع أنوعها.

بالنسبة إلى الخرج فهو المكان الذى يتم فيه توصيل طرف واحد فقط من طرفى الحمل بحيث يقوم باستقبال الإشارة **من** الــ PLC وهذا يطبق على الأحمال ذات الإشارة الرقمية بجميع أنوعها.



ملاحظة:

- يتم توصيل الطرف الأخر من المفاتيح بالتيار الكهربى مع مراعاة قيمة الڤولت المكتوب على جهاز الــ PLC.
- يتم توصيل الطرف الأخر من الأحمال بالسالب مع مراعاة أن جميع أجهزة الــ PLC تعطى إشارة موجبة فى الخرج ولذالك يتم توصيل الطرف الأخر بالسالب.

تنقسم المدخلات إلى مداخل ذات إشارة رقمية و أخرى ذات أشاره تناظرية و تنقسم المخرجات أيضاً إلى مخارج ذات إشارة رقمية و أخرى ذات أشاره تناظرية.

أنواع المداخل والمخارج والفرق بينهما

الإشارة الرقمية:

- **مداخل رقمية:**

توصل المداخل الرقمية بوحدات المداخل الرقمية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الدخل الرقمية بالتحكم فى أشارات المداخل تبعاً لظروف التشغيل (وعادة فإن قيمة الكهربية يكون لها قيمتين فقط: واحد أو صفر، وتتغير القيمة الكهربية من و إلى هاتين القيمتين حسب حالة المفتاح.

- **مخارج رقمية:**

توصل المخارج الرقمية بوحدات الخرج الرقمية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الخرج الرقمية بالتحكم بالمخارج، وجميع هذه المخارج لها حالتين فقط: حالة تشغيل، وحالة توقف.

الإشارة التناظرية.

- **مداخل تناظرية:**

توصل أجهزة المداخل التناظرية بوحدات المداخل التناظرية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الدخل التناظرية بتحويل أى كمية مطلوب قياسها إلى كمية كهربية مثل الجهد والتيار (وعادة فإن القيمة الكهربية يكون لها قيمتين: قيمة عظمى وقيمة صغرى، وتتغير القيمة الكهربية بين هاتين القيمتين.

- **مخارج تناظرية:**

توصل أجهزة المخارج التناظرية بوحدات الخرج التناظرية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة المخارج التناظرية بالتعامل مع القيم المتغيرة الناتجة عن وحدات الخرج الخاصة بوحدة التحكم المنطقى وجميع هذه الأجهزة لها أكثر من حاله: وعادة فإن حالة الخرج تتغير حسب القيمة الكهربية للخرج التناظرى و تتغير القيمة الكهربية بين قيمة عظمى وقيمة صغرى.

أمثلة لمداخل ذات إشارة رقمية.

١– الضواغط اليدوية Push Buttons:

وهذه الأجهزة ينعكس حالة ريش تلامسها أى تصبح الريشة المفتوحة طبيعياً مغلقة، والعكس صحيح، وذلك عند الضغط على رؤوسها.



٢– مفاتيح نهاية المشوار Limit Switches :

وهذه الأجهزة يتغير حالة تلامسها عند دفع عنصر الفعل لها بكامة متحركة. توجد أنواع مختلفة لمفاتيح نهاية المشوار الميكانيكية.



٣– المفاتيح التقاربية Proximity Switches :

وهذه الأجهزة ينعكس وضع ريشة تلامسها عند اقتراب جسم غريب منها لمسافة معينة تعتمد على مدى تشغيل المفتاح التقاربى





٤– مفاتيح الخلايا الضوئية Photocell Switches :

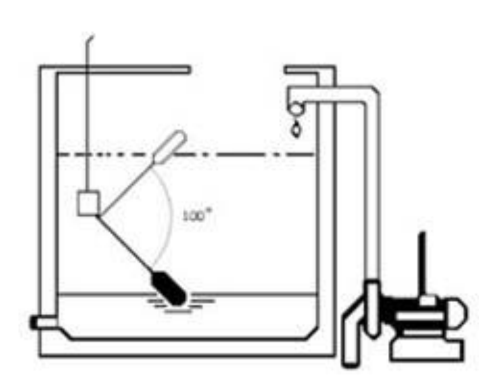
وينعكس حالة ريش تلامس هذه الأجهزة عند مرور جسم غريب ليقطع الشعاع المنبعث من وحدة الإرسال للخلية الضوئية, أى أنه عند مرور أى جسم غريب بين المستقبل والمرسل للخلايا الضوئية تتغير نقط التلامس الخاصة بالخلايا الضوئية.





٥- مفاتيح العوامات Float Switches :

وتستخدم هذه لمفاتيح لتتبع مستوى السوائل فى الخزانات حيث ينعكس حالة ريش هذه المفاتيح عند وصول مستوى السائل فى الخزانات إلى مستواها، فيستخدم فى تتبع مستوى السوائل فى الخزانات.



٦- مفاتيح الضغط Pressure Switches:

وهى أجهزة ينعكس حالة ريش تلامسها عند وصول الضغط فى الأنابيب والخزانات إلى الضغط المضبط مسبقاً، وتستخدم هذه المفاتيح لمتابعة ضغوط السوائل والغازات.



٧- مفاتيح درجة الحرارة Thermo states :

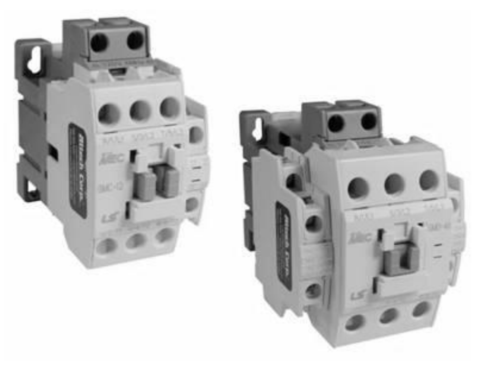
وهى أجهزة ينعكس حالة ريش تلامسها عند ارتفاع درجة حرارة الوسط المحيط لعنصر إحساسها وصولاً لدرجة الحرارة المعايرة عليها هذه المفاتيح.



أمثلة لمخرجات ذات إشارة رقمية.

١- الكونتاكتورات Contactors :

ويقوم الكونتاكتور بتوصيل التيار الكهربى إلى الأحمال عند وصول جهد كهربى إلى ملفه والعكس صحيح. يتكون الكونتاكتور من ملف كهربى (البوبينة) وقلب مغناطيسى له شق ثابت، وآخر متحرك يحمل ريش تلامس رئيسية، وعند وصول جهد كهربى على أطراف ملف الكونتاكتور ينجذب الشق المتحرك للقلب تجاه الشق الثابت، فتنعكس ريش الكونتاكتور.



٢- **الريليهات الأستاتيكية Static relays** :

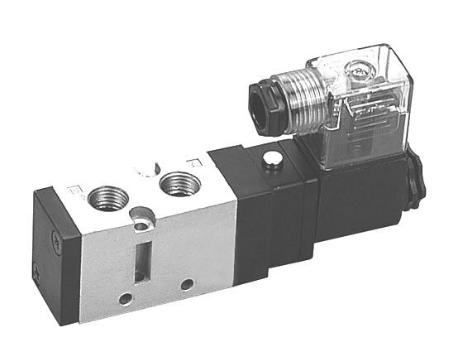
وتقوم بتوصل وفصل التيار الكهربى عن الأحمال - تماماً مثل الكونتاكتورات - ويفضل استخدام ها بدلاً من الكونتاكتورات عند زيادة عدد مرات التوصيل والفصل فى الدقيقة.



٣- **المحابس الكهربية Solenoid Valves** :

تقوم بفتح أو غلق مسارات مرور السوائل فى الأنابيب وتتكون من: ملف كهربى، وقلب مغناطيسى ثابت وقلب مغناطيسى متحرك، يقوم بفتح أو غلق المحبس وعند وصول تيار كهربى لملف المحبس الكهربى يتحرك الجزء المتحرك للقلب المغناطيسى فيفتح مسار مرور السوائل وهكذا ...





٤– لمبات البيان **Indication Lamps** :

وهى توجد فى غرف التحكم لمساعدة المشغلين على فهم أداء العمليات الصناعية و توضيح الأعطال.



٥– الأبواق **Horns** :

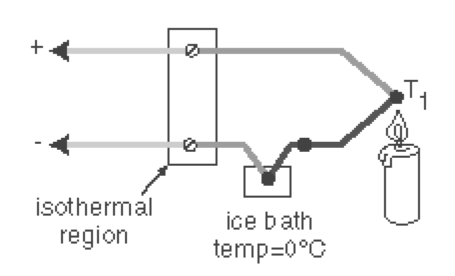
وهى تصدر أصوات عالية عند حدوث أمر غير عادى فى العملية الصناعية لتنبيه المشغلين، وتعمل عند وصول تيار كهربى لملفها, حيث أنها تعتمد على تدفق الهواء المضغوط داخل زور البوق.

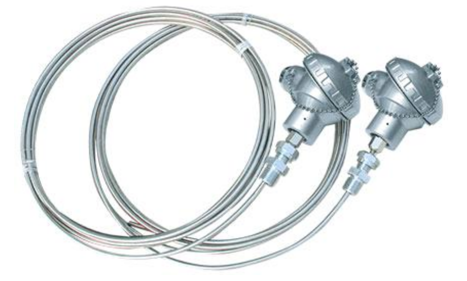


أمثلة لمداخل ذات إشارة تناظرية.

١- الازدواج الحرارى Thermo Couple :

يقوم الازدواج الحرارى بتحويل درجة الحرارة إلى إشارة جهد ويتكون الازدواج الحرارى من معدنين مختلفين (أ - ب) متصلين معاً لتكوين مجس القياس، وتعتمد قيمة الجهد المتولد على درجة حرارة الوصلة, و لذلك فأن الجهد على أطراف الأذدواج الحرارى يتناسب طردياً مع درجة الحرارة.





٢- مولد التاكو Tacho generator :

وهذا المولد يثبت على أعمدة المحركات المطلوب قياس سرعتها، وخرج مولد التاكو خطى،بمعنى أن جهد أطرافه يتناسب طردياً مع السرعة، فإذا كانت نسبة التحويل المولد التاكو هى واحد فولت لكل **٣٠٠** لفة فى الدقيقة الواحدة، فمثلاً إذا كان الجهد على طرف المولد هو ٥ فولت، يعنى هذا أن سرعة المحرك هى ١٥٠٠ لفة فى الدقيقة الواحدة و هكذا ...



أمثلة لمخارج ذات إشارة تناظرية.

١– الدرايفر Driver :

يتم استخدام الدرايفر كمثال للخرج التناظرى حيث يتم استقبال قيمة كهربية (mv – ma – v – Ω) من وحدة البرمجة المنطقية وبواسطة جهاز الدرايفر يتم التحكم فى سرعة الموتور عن طريق تغيير قيمة التردد.



٢– محرك الدامبر Damper motor:

ينم استخدام المحرك الدامبر كمثال للخرج التناظرى حيث يتم استقبال قيمة كهربية (mv – ma – v – Ω) من وحدة البرمجة المنطقية وبواسطة المحرك الدامبر يتم التحكم فى درجة التبريد الخاصة بأجهزة التكيف المركزية حيث يقو بفتح بوابة مرور الهواء البارد حسب الاحتياج.



٤- الذاكرة..........**memory**

الذاكرة داخل الـــ PLC مهمة جداً لأنها تحتوى على البرنامج الذى يقوم الـــ CPU بقراءته ولهذا من الممكن فصل الكمبيوتر عن الـــ PLC بعد تحميل البرنامج على الذاكرة.



ملاحظة:

- إذا كانت الذاكرة التى تحتوى على البرنامج من النوع المتطاير...volatile memory كما سيوضح فيما بعد فيجب أن يكون جهاز الـــ PLC موصل بمصدر مستمر للتيار مثل (battery, plug) لحفظ البرنامج.

- إذا كانت الذاكرة التى تحتوى على البرنامج من النوع غير المتطاير... permanent memory فليس من الضرورى أن يكون جهاز الـــ PLC موصل بمصدر مستمر للتيار.

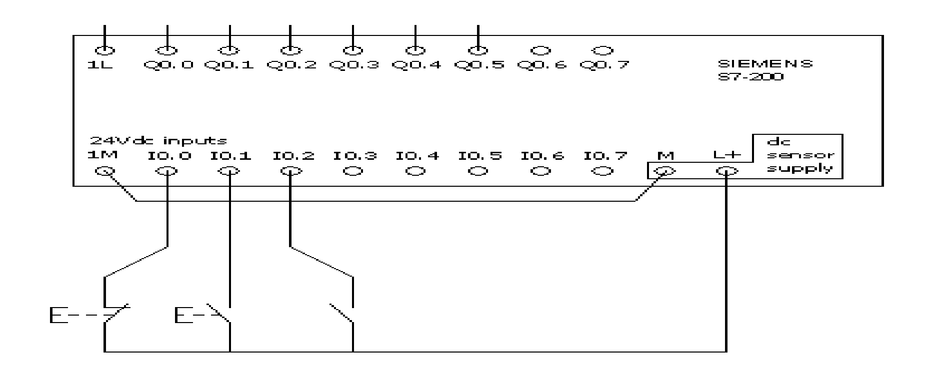
- لا يمكن تخزين أكثر من برنامج على الذاكرة فى نفس الوقت.

تصنيف وحدة الـــ PLC

١– الترانزستور الثنائي القطب...bipolar junction transistor:

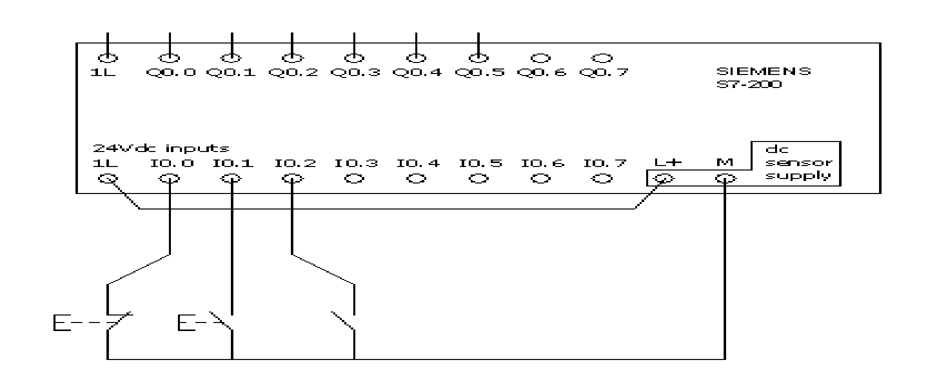
- PNP(Positive Negative Positive):

إذا كان الـــ PLC من النوع PNP يتم تغذية جميع المفاتيح بإشارة موجبة بينما يتم توصيل السالب على وحدة الدخل حيث يوجد حرف M كما هو موضح بالرسم.



- NPN(Negative Positive Negative):

إذا كان الـــ PLC من النوع NPN يتم تغزيه جميع المفاتيح بإشارة سالبة بينما يتم توصيل الموجب على وحدة الدخل حيث يوجد حرف L+ كما هو موضح بالرسم.



ملاحظة:

أ - يفضل أن يكون الــ PLC من نوع NPN وليس PNP لأن فى حالة استخدام PLC (PNP) يكون الطرف الموجب متصل بالمفاتيح لذلك إذا قام العامل بلمس الطرف الموجب بالخطأ بينما تلمس قدميه الأرض سوف يصاب العامل بصدمة كهربية, أما إذا كان الــ PLC (NPN) يكون الطرف السالب متصل بالمفاتيح لذالك إذا قام العامل بلمس الطرف السالب بالخطأ و تلمس قدميه الأرض لن يصاب العامل بصدمة كهربية قط لأن فرق الجهد بين السالب و الأرض يساوى صفر.

ب -نوع الــ PLC سواء كان PNP أو NPN فهذا يشير فقط إلى طريقة توصيل الدخل input وليس له علاقة بطريقة توصيل الخرج output لأن جهاز الــ PLC يعطى إشارة للخرج موجبة دائماً ,كما ذكرت سابقاً, بغض النظر عن نوع جهاز الــ PLC .

ت -حرف الــ L المكتوب على جهاز الــ PLC يعنى مكان توصيل الطرف الموجب بينما حرف الــ M يعنى مكان توصيل الطرف السالب وهذا أن دل على شيء فأنة يدل على أن هذا النوع من أجهزة الــ PLC تعمل بالتيار المستمر.

ث -يعتمد أيضاً استخدام نوع الــ PLC حسب النوع المتوافر فى الأسواق ولذالك فأنها تتغير من قارة إلى قارة أو من بلد لأخرى.

٢- الإشارة الرقمية و الإشارة التناظرية.........Digital & Analog

أ –الإشارة الرقمية.....Digital:

المقصود بالإشارة الرقمية digital هى أى إشارة رقمية لها حالتين فقط, أما أن تساوى الإشارة واحد أما أنها تساوى صفر.

فمثلاً إذا تكلمنا عن digital input فالحالتين هم:

أن كان المفتاح مغلق (YES – TRUE – ON) ويرمز لها بواحد (١).

أن كان المفتاح مفتوح (NO – FALSE – OFF) ويرمز لها بصفر (٠).

وبالمثل إذا تكلمنا عن digital output فالحالتين هم:

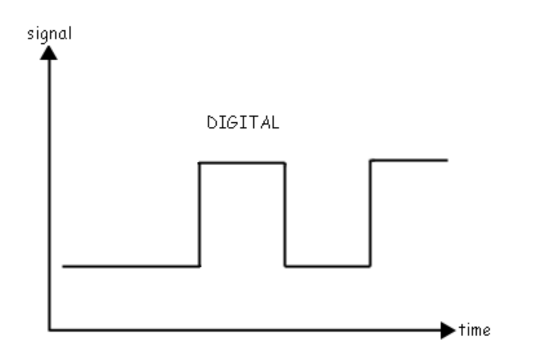
أن كان الخرج يعمل (YES – TRUE – ON) ويرمز لها بواحد (١).

أن كان الخرج لا يعمل (NO – FALSE – OFF) ويرمز لها بصفر (٠).



- مثال للمداخل الرقمية: (مفتاح عادى – مفتاح جرس – مفتاح نهاية المشوار).
- مثال للمخارج الرقمية: (لمبة – جرس – موتور – مضخة).

شكل الإشارة الرقمية:



ب –**الإشارة التناظرية**.....Analog:

المقصود بالإشارة التناظرية Analog هى أى إشارة لها أكثر من حالتين أى أن الإشارة لها قيم متغيرة بخلاف الصفر.

فمثلاً إذا تكلمنا عن Analog input:

أن كانت توجد إشارة, فقد تكون: (١-٢-٣-٤-......-٣٢٧٦٧).

أن كانت لا توجد إشارة: (صفر).

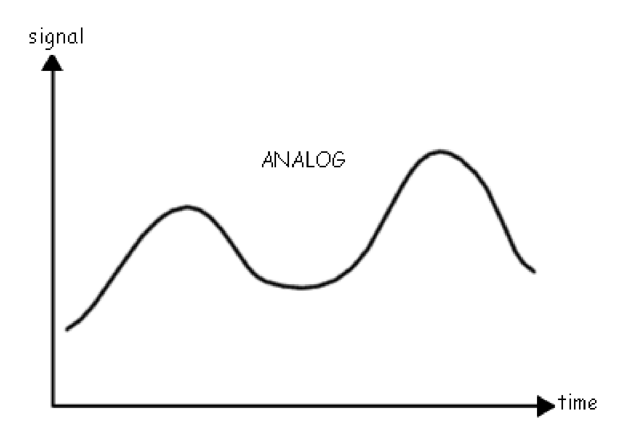
وبالمثل إذا تكلمنا عن Analog output:

أن كانت توجد إشارة, فقد تكون: (١-٢-٣-٤-......-٣٢٧٦٧).

أن كانت لا توجد إشارة: (صفر).



- مثال للمداخل التناظرية: (ترمومتر – المقاومة المتغيرة - إنكودر).
- مثال للمخارج الرقمية: (سخان – الفولتميتر).



ملاحظة:

- رقم ٣٢٧٦٧ هو رقم ناتج عن مساحة الـــ word.

٣- أنواع المخرجات: (ريليه أو ترانزستور) ... Transistor or Relay

النوع الأول: ترانزستور..... Transistor

إذا كان نوع الـــ PLC هو output transistor فهذا لا يجعل طريقة التوصيل تختلف ولكن له بعض المميزات والعيوب مقارنتاً بالنوع الأخر (output relay):

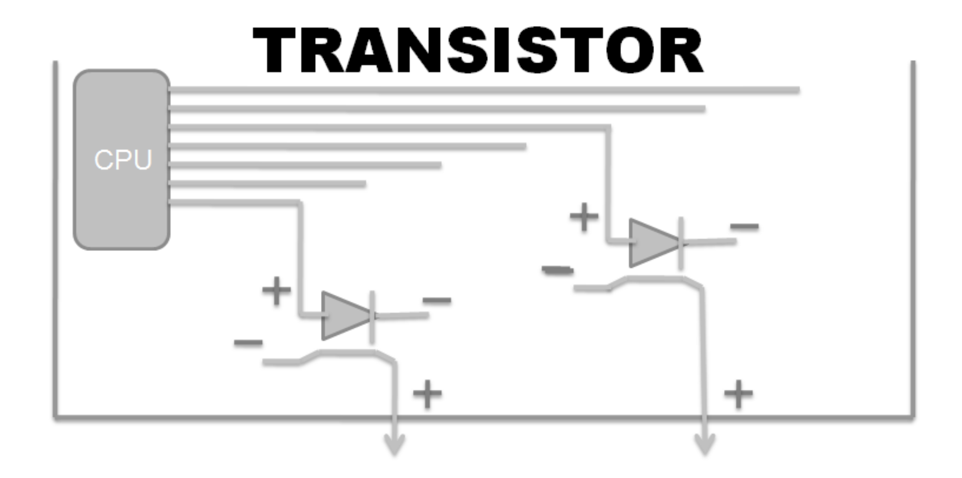
المميزات:

١- يمكنه إرسال إشارات سريعة فى وقت قصير حيث أنة قد يرسل أكثر من ألف إشارة فى الثانية تقريباً.

العيوب:

١- القولت الخاص بالـــ output ثابت وقيمته تساوى 24V DC.

٢- لا يتحمل تيار أكثر من 0.36A – 0.5A.



النوع الثانى: ريليه.......Relay

إذا كان نوع الــ PLC هو output relay فهذا لا يجعل طريقة التوصيل تختلف ولكن له بعض المميزات والعيوب مقارنتاً بالنوع الأخر (output transistor):

المميزات:

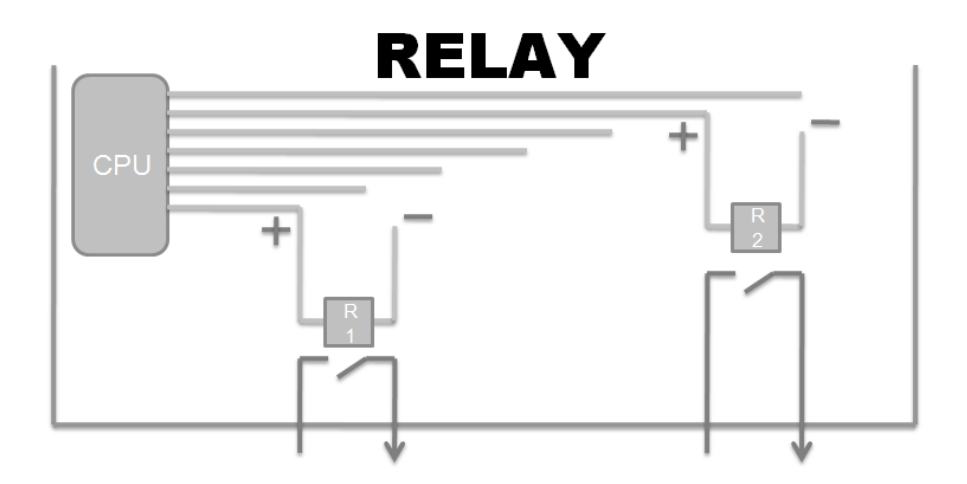
١- الڤولت الخاص بالــ output غير محدد بل من الممكن توصيل أى قيمة ضمن الحد المسموح به.

مثلاً: 12V DC , 110VAC , 220V AC, 24V DC.

٢- يتحمل تيار يصل إلى 2A - 2.5A.

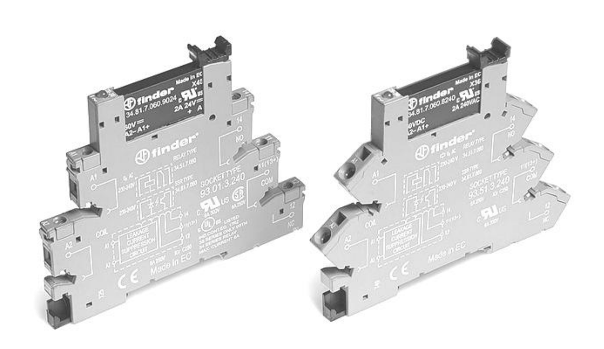
العيوب:

١- لا يمكنه إرسال إشارات سريعة فى وقت قصير مثل الترانزستور.



ملاحظة:

١- لحل العيوب الموجودة بالنوعين, سواء فى الــ **output transistor** أو **output relay** يتم استخدام ريليه ميكانيكى **relay interface**.



الريليه الميكانيكى......Relay interface

الريليه الميكانيكى يعمل كوسيط بين الــ PLC و الحمل حيث يقوم الــ PLC بتشغيل الريليه بينما يقوم بعد ذلك الريليه بتشغيل الحمل المراد تشغيله بواسطة استخدام نقاط المساعدة.

يتم شراء الريليه على أساس مواصفات تخص الملف coil و مواصفات أخرى تخص النقاط contact.

أولاً: مواصفات الملف...Coil:

١- أن يكون الجهد الخاص بملف الريليه يعمل بنفس قيمة الجهد الخارج من وحدة الــ PLC.

٢- أن يكون التيار المسحوب من ملف الريليه ضمن الحد المسموح به لكى لا يضر بوحدة الــ PLC.

ثانياً: مواصفات نقاط التلامس...Contact:

١- أن تتحمل نقاط الريليه الجهد الخاص بالحمل الذى سيعمل بواسطة النقاط المساعدة.

٢- أن تتحمل نقاط الريليه قيمة التيار المسحوب من الحمل الذى سيعمل بواسطة هذا الريليه.

لماذا يستخدم الريليه؟

١- للتمكن من تشغيل الأحمال التى تعمل بڤولت مختلف عن الڤولت الخارج من جهاز الــ PLC.

٢- للتمكن من تشغيل الأحمال التى تسحب تيار بقيمة أكبر من التى يتحملها جهاز الــ PLC.

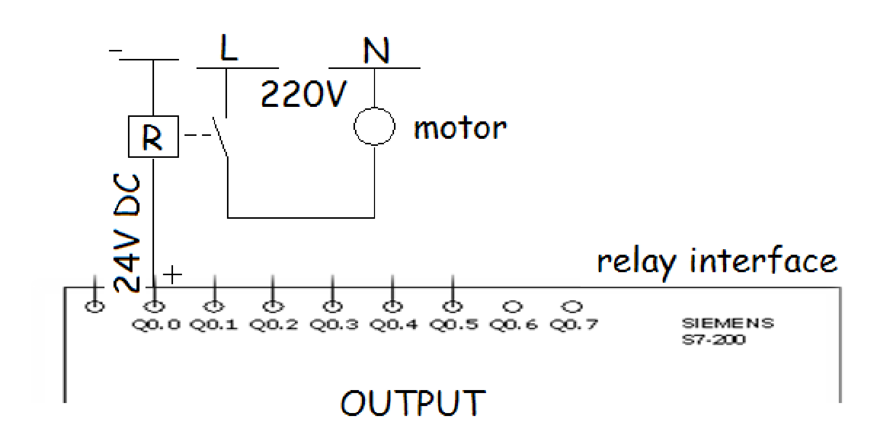
٣- لحماية جهاز الــ PLC من التيار الزائد الذى قد يسحبه الحمل فى أى وقت.

متى يستخدم الريليه؟

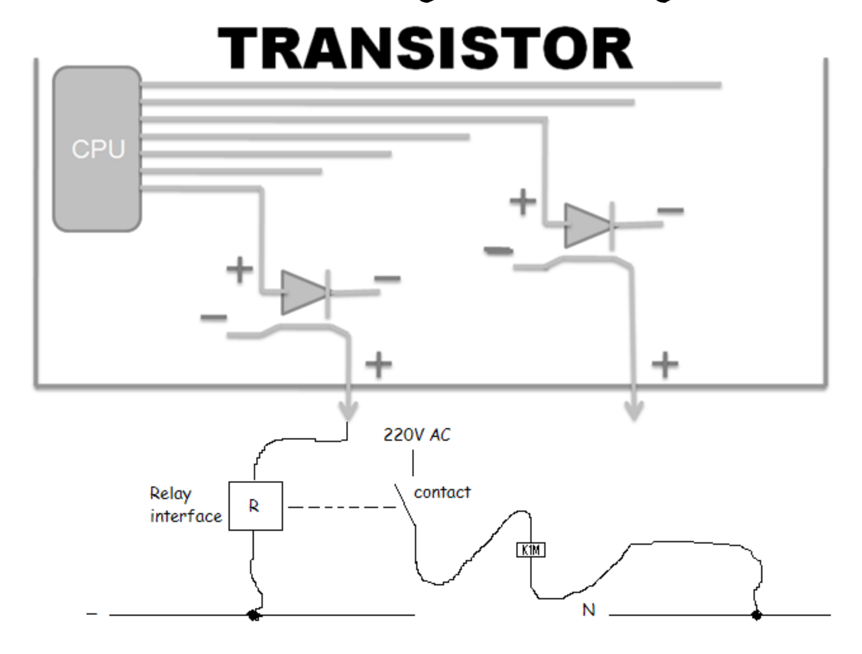
١- إذا كانت الأحمال المستخدمة تختلف فى التيار أو فى الجهد مع جهاز الــ PLC.

٢- وأيضا إذا كان الحمل يتوافق مع الــ PLC من حيث التيار و الجهد ولكن من الممكن إذا كان هذا الحمل محرك أن يسحب تيار زائد لزيادة قوة العزم مثلاً.

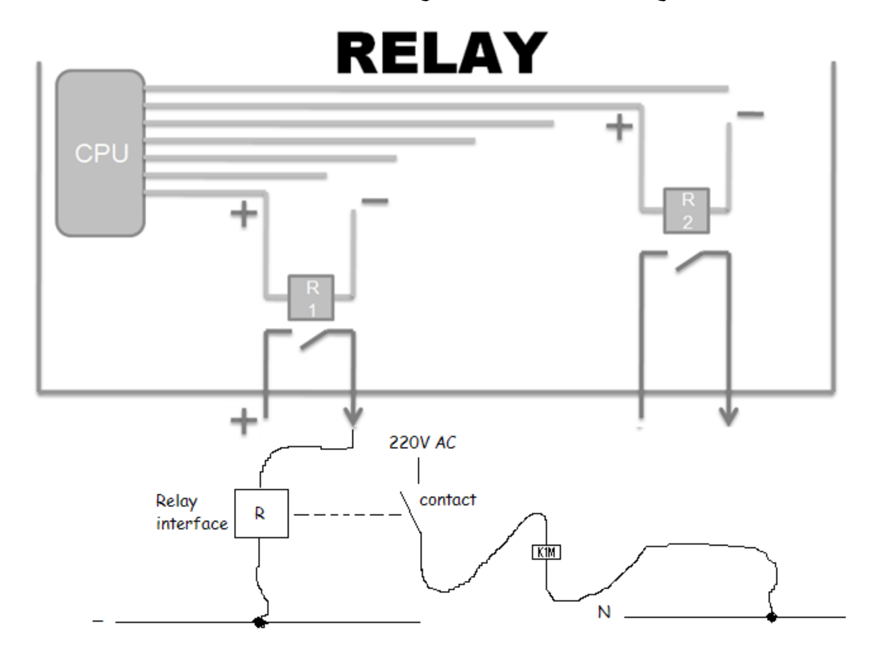
طريقة توصيل الريليه



- طريقة توصيل أحمال مع وحدة **PLC** من نوع الـــ **output transistor**:



- طريقة توصيل أحمال مع وحدة **PLC** من نوع الـــ **output relay**:



ملاحظة:

١- فى حالة استخدام الريليه يصبح من الممكن تغذية أى حمل يعمل بأى جهد ويسحب أى تيار فى حدود نوع الريليه المستخدم.

٢- إذا سحب الموتور تيار ذائد فليس هناك أى خطر على الـــ PLC لأنه سوف يتأثر الريليه وليس الـــ PLC.

٣- أختار الريليه بحيث أن يعمل بنفس الڤولت الخارج من الـــ PLC بينما أختار نقاط الريليه بحيث تتحمل التيار المسحوب من الحمل.

٤- من المحتمل أن الحمل يسحب تيار ذائد من نقطة الريليه و لكن ليس من الممكن أن يسحب الريليه تيار ذائد من الـــ PLC لأن الريليه ليس إلا ملف solenoid.

٥- قد يجد البعض مشكلة فى استخدام الريليه الميكانيكى وهذا لأنه يتكون من ملف و نقط تلامس تعمل ميكانكياً فلذلك قد يتطلب وقت بين الإشارات حتى تقوم نقط التلامس بالفتح و بالغلق ولذلك فإذا كان الـــ PLC يصدر أشارات سريعة فيفضل إذاً استخدام الريليه الإلكترونى solid state relay عوض عن الريليه الميكانيكى حيث أنه يتميز بنقاط مساعدة تقوم بتغير الحالة بسرعة فائقة .

كيفية توصيل وحدة الـــ PLC بالكامل:

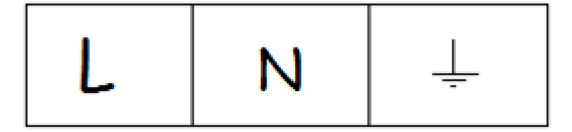
ينقسم التوصيل إلى ثلاث أجزاء:

(تغذية وحدة الـــ CPU – تغذية مجموعة المدخلات – تغذية مجموعة المخرجات)

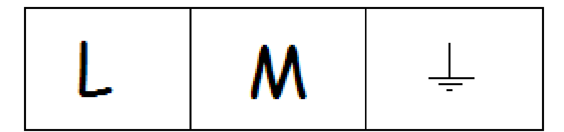
١– تغذية الـــ CPU.

تغذية الـــ CPU تطلب فقط توصيل الكهرباء حسب نوع الـــ PLC (مثلاً التيار المتردد أو التيار المستمر).

تيار متردد



تيار مستمر

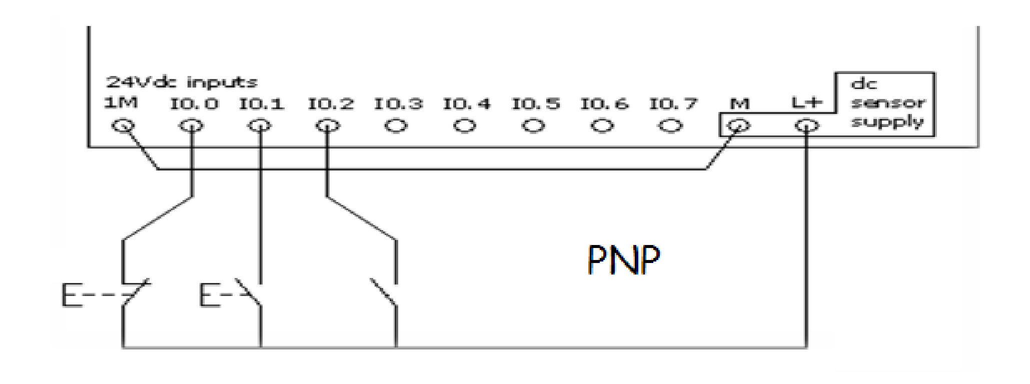


٢– تغذية مجموعة المدخلات.

كما سبق وشرحنا, تغذية مجموعة المدخلات تتطلب أولاً توصيل طرف من الكهرباء على المفاتيح وثانياً توصيل الطرف الأخر على وحدة المدخلات, قد يختلف التوصيل قليلاً حسب نوع الـــ PLC (PNP أو NPN).

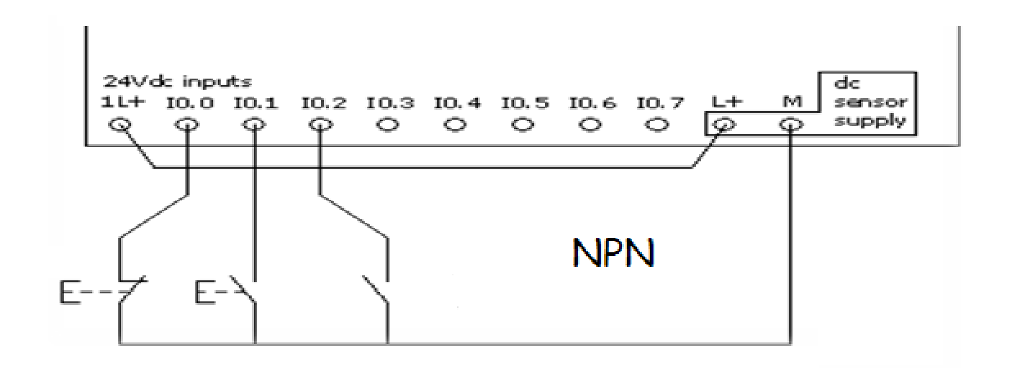
النوع الأول. PNP:

يتم توصيل الطرف الموجب على المفاتيح بينما يوصل الطرف السالب على وحدة الدخل نفسها.



النوع الثانى. NPN

يتم توصيل الطرف السالب على المفاتيح بينما يوصل الطرف الموجب على وحدة الدخل نفسها.



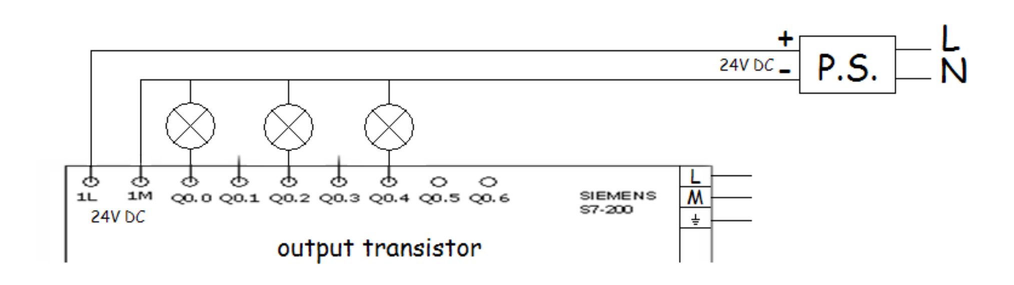
٣– تغذية مجموعة المخارج.

تغذية مجموعة المخرجات تتطلب فقط توصيل الكهرباء حسب نوع الــ PLC (مثلاً transistor أو relay).

النوع الأول. Transistor:

فى حالة الــ output transistor يتم استخدام التيار المستمر فقط ولذلك يعوض عن كلمة transistor بكلمة DC

ودائماً ما يكون الجهد الخاص بالخرج هو 24V DC كما هو مبين فر الرسم.



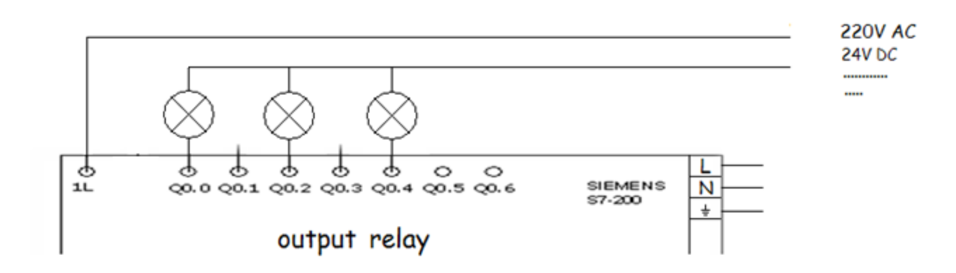
النوع الثانى. Relay:

فى حالة الــ output relay يتم استخدام التيار المستمر أو التيار المتردد, يعوض عن كلمة relay بكلمة RLY

الجهد الخاص بالخرج هو متغير حيث يمكن توصيل أى قيمة ضمن الحد المسموح به من جهاز الــ PLC.

مثلاً: (24V DC – 220V AC - ……..V DC - …….V AC)

كما هو مبين بالرسم.



كيف أعرف نوع الـــ PLC.

يوجد نوعين من حيث طريقة التوصيل:

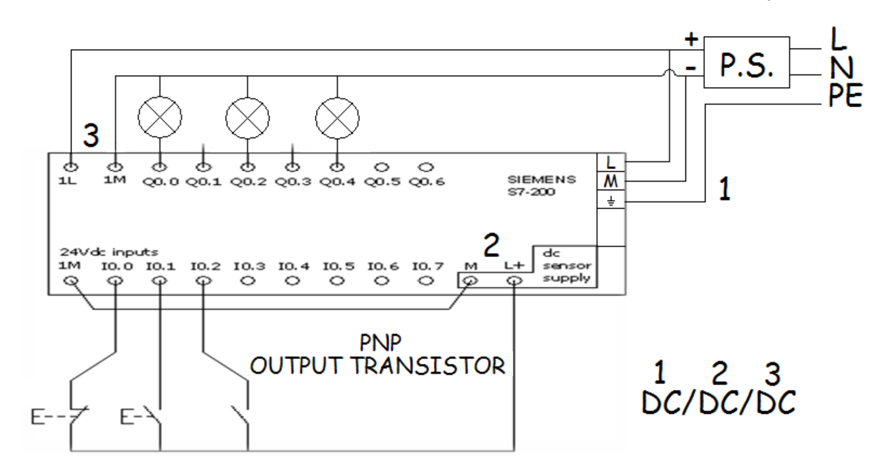
<u>DC/DC/DC</u> - **<u>AC/DC/RLY</u>**

النوع الأول. DC/DC/DC:

- كلمة **DC** الأولى تخص تغذية الـــ CPU.
- كلمة DC الثانية تخص تغذية المداخل.
- كلمة DC الثالثة تخص تغذية المخارج.

المقصود بـــ <u>**DC** الأولى</u> هى كلمة **DC** التى فى أقصى اليسار, أنظر الشكل التالى.

رسم توضيحى للنوع الأول **DC/DC/DC**.



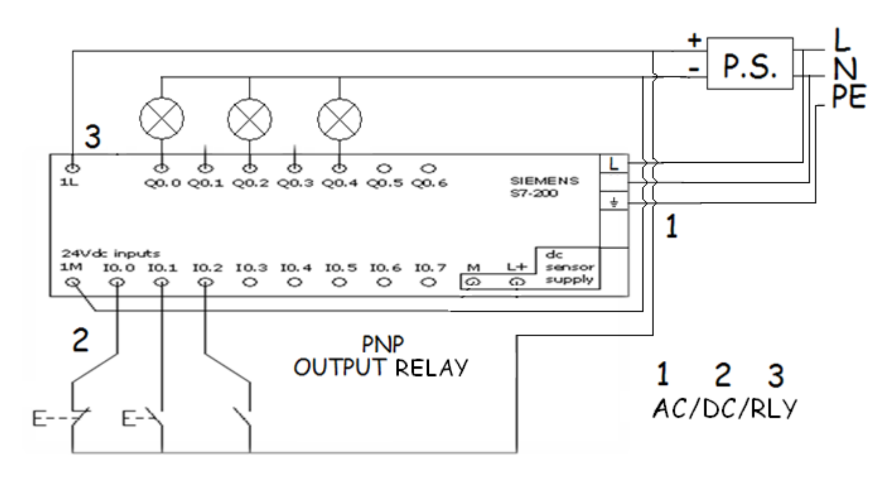
النوع الثانى. **AC/DC/RLY**:

كلمة **AC** تخص تغذية الـــ CPU.

كلمة **DC** تخص تغذية المداخل.

كلمة **RLY** تخص تغذية المخارج.

رسم توضيحى للنوع الثانى.



ملاحظة:

- جميع الموديلات يتوفر منها النوعين: DC/DC/DC و AC/DC/RLY.
- الـ output transistor هو DC/DC/DC.
- الـ output relay هو AC/DC/RLY.

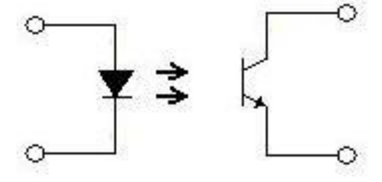
الأمثال للنوعين:

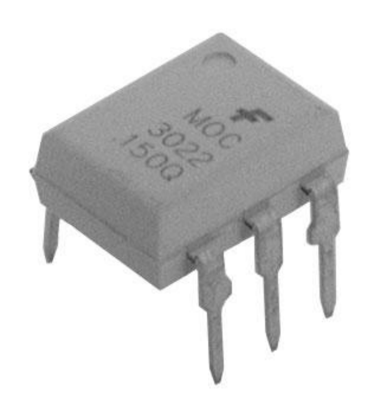
Model Description	Power Supply	Input Types	Output Types
221 DC/DC/DC 221 AC/DC/Relay	20.4-28.8 VDC 85-264 VAC 47-63 Hz	6 DC Inputs 6 DC Inputs	4 DC Outputs 4 Relay Outputs
222 DC/DC/DC 222 AC/DC/Relay	20.4-28.8 VDC 85-264 VAC 47-63 Hz	8 DC Inputs 8 DC Inputs	6 DC Outputs 6 Relay Outputs
224 DC/DC/DC 224 AC/DC/Relay	20.4-28.8 VDC 85-264 VAC 47-63 Hz	14 DC Inputs 14 DC Inputs	10 DC Outputs 10 Relay Outputs
226 DC/DC/DC 226 AC/DC/Relay	20.4-28.8 VDC 85-264 VAC 47-63 Hz	24 DC Inputs 24 DC Inputs	16 DC Outputs 15 Relay Outputs

حمايات داخلية للـــ PLC .

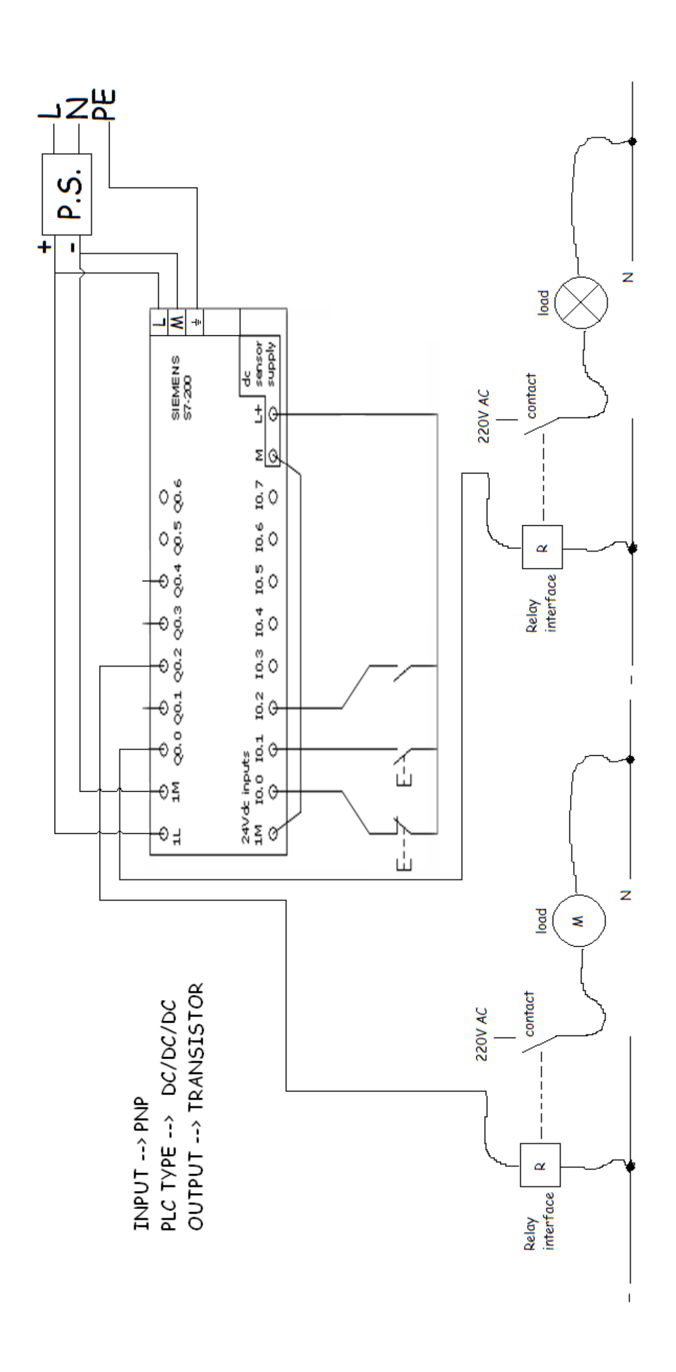
عازل كهروضوئى

يوجد داخل جهاز الــ PLC عازل كهروضوئى optical coupler خاص بكل دخل (input) بحيث فى حالة توصيل أى جهد زيادة عن طريق الخطاء فأنه فى الغالب سوف يتضرر العازل الكهروضوئى وليس الــ PLC وذالك لأن العازل الكهروضوئى يستعمل كعازل بين دائرتين كهربائيتين, حيث الإشارة الكهربائية القادمة من الدارة الأولى تتحول إلى إشارة ضوئية ومن ثم تقوم الدارة الثانية بتحويل هذا الإشارة الضوئية إلى كهربائية مرة أخرى, يعمل هذا العازل مع الإشارات الرقمية وكذلك مع التماثلية أو التناظرية .





الشكل العام للـــ PLC :



بالنظر لجهاز الــ PLC سوف يلاحظ:

١- لمبات إشارة.

٢- مفتاح التحكم بجهاز الــ PLC.

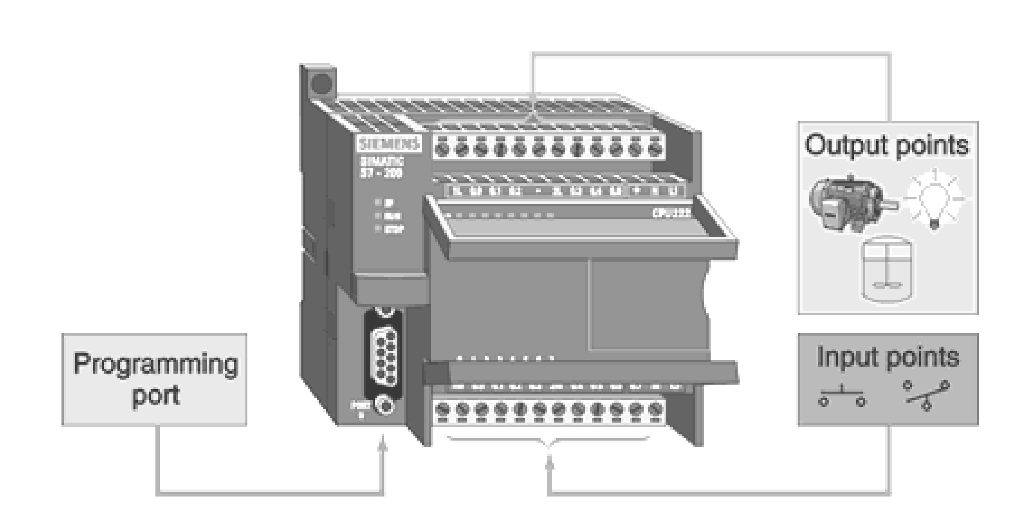
٣- مفتاح الضبط التناظرى Analog adjustment.

٤- مدخل كابل البرمجة.

٥- مكان البطارية.

٦- مكان الذاكرة.

٧- مدخل وحدة دخل أو خرج إضافية Extension module.



١– لمبات الإشارة.....**Indicators**.

تقسم إلى ثلاث لمبات:

Run - Stop - S.F.

- لمبة **RUN**:

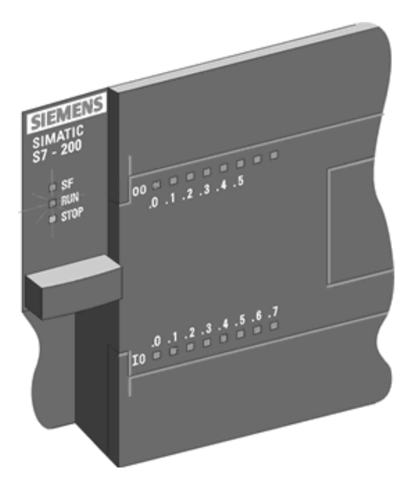
لمبة run تضاء لون أخضر حينما يعمل الـــ PLC, أى عندما يكون الجهاز فى وضع run.

- لمبة STOP:

لمبة stop تضاء لون أخضر حينما يقف الـــ PLC أى عندما يكون الجهاز فى وضع stop.

- لمبة SYSTEM FAULT:

لمبة system fault تضاء لون أحمر حينما يقف الـــ PLC بسبب مشكلة ما.



ملاحظة:

فى حالة أن لمبة الـــ .S.F مضاءة من المؤكد أن لمبة الـــ stop ستكون مضاءة أيضاً نظراً لأن فى حالة وجود مشكلة فى الـــ PLC ليس من الممكن أن يستمر فى العمل أى أنه ليس من المنطقى أن يبقى فى وضع run.

٢- **مفتاح التحكم بجهاز الــ PLC......Mode Switch**:

توجد ثلاث أوضاع لمفتاح التحكم بجهاز الــ PLC

Run - Stop - Terminal

- RUN:

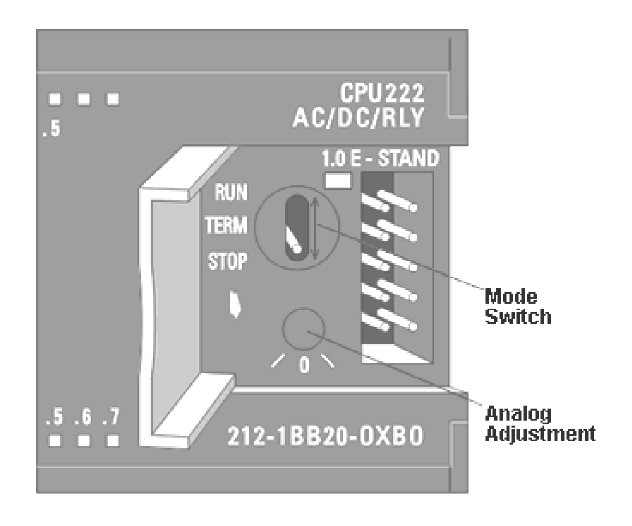
في حالة ضبط المفتاح على وضعية run يعمل جهاز الــ PLC أوتوماتيكياً وتضئ لمبة run باللون الأخضر.

- STOP:

في حالة ضبط المفتاح على وضعية stop يقف جهاز الــ PLC أوتوماتيكياً وتضئ لمبة stop باللون الأخضر.

- TERMINAL:

في حالة ضبط المفتاح على وضعية terminal يصبح من الممكن التحكم فى جهاز الــ PLC أوتوماتيكياً عن طريق الكمبيوتر وتضئ لمبة stop أو لمبة run حسب الأختيار.



٣– مفتاح ضبط الإشارة التناظرية.....Analog adjustment:

مفتاح ضبط الإشارة التناظرية يستخدم كمثال للدخل التناظرى حيث من الممكن تغير قيمة المفتاح بواسطة مفك لكى يستخدم فى البرمجة (يمكن تغير القيمة من صفر إلى ٢٥٥).

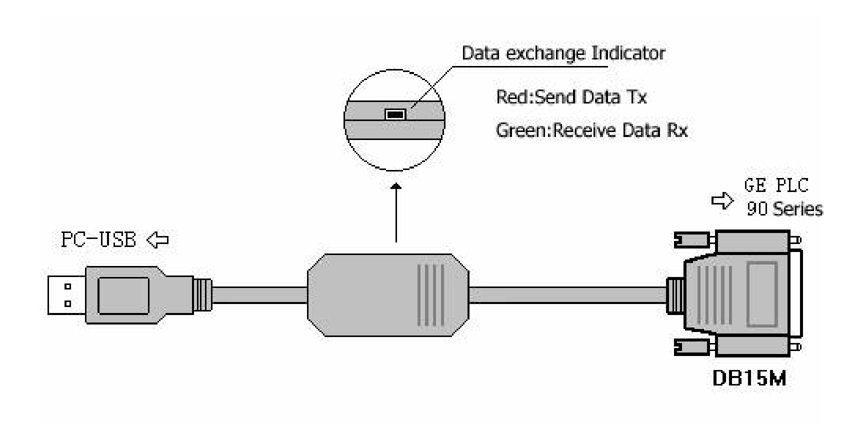
٤- مدخل كابل البرمجة.........Programming Cable:

حيث يتم توصيل الكابل بين جهاز الـــ PLC و جهاز البرمجة وهو الكمبيوتر فى هذه الحالة. و يسمى التوصيل بينهم بأسم PPI (Point to Point Interface) ويستخدم الكابل لنقل المعلومات من و إلى الـــ PLC.

يختلف نوع الكابل من بين بعض الأجهزة الموجودة بالسوق:

١ - كابل ذات سرعة 9.6Kbps
٢- كابل ذات سرعة 19.2Kbps
٣ -كابل ذات سرعة 187.5Kbps

-الكابل ذات سرعة 9.6Kbps يستطيع أن ينقل معلومات بحجم 9.6Kb فى ثانية واحدة فقط.
-الكابل ذات سرعة 19.2Kbps يستطيع أن ينقل معلومات بحجم 19.2Kb فى ثانية واحدة فقط.
وهكذا......



ملاحظة:

الكابل الخاص بالـ S7 200 توجد به ثلاث لمبات إشارة (Data exchange Indicator):

PPI - Tx - Rx

-اللمبة الأولى PPI.

تسمى PPI وهي أختصار لكلمة Point to Point Interface وهى تضاء بصفة مستمرة فى حالة أتصال جهاز الـ PLC بجهاز التحكم (computer) وتضاً بالون الأصفر.

-اللمبة الثانية Tx.

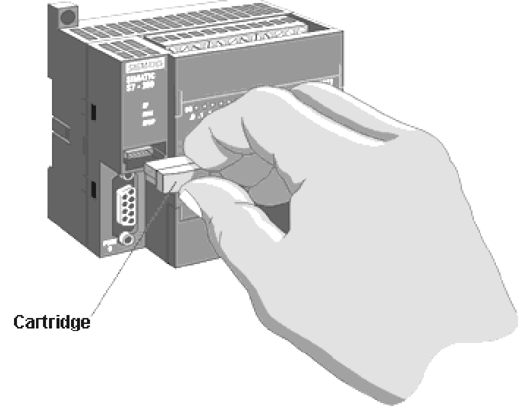
تسمى Tx وهي أختصار لكلمة Transfer Data وهى تضاء بصفة متقطعة أثناء أرسال معلومات إلى جهاز الـ PLC وتضاً بالون الأصفر.

-اللمبة الثالثة Rx.

تسمى Rx وهي أختصار لكلمة Receive Data وهى تضاء بصفة متقطعة أثناء إستقبال معلومات من جهاز الـ PLC وتضاً بالون الأصفر.

٥- البطارية.......Battery:

يتم توصيل البطارية في جهاز الـ PLC لكي تعمل كمصدر مستمر للتيار لهدف الحفاظ على البرنامج داخل الذاكرة حتى فى حالة انقطاع المصدر الرئيسى للتغذية (الكهرباء).



ملاحظة:

من أكثر أنواع البطاريات المستخدمة هو Lithium:

- جهد البطارية هو 3.6V
- يفضل تغيير البطارية كل سنتين.

المميزات:

- التيار (ampere curve) الخاص بالبطارية ثابت.

العيوب:

- التيار (ampere curve) الخاص بالبطارية يتزل إلى صفر عند نهاية العمر الإفتراضى للبطارية دون سابق أنظار.



٦- الذاكرة.......Memory:

تنقسم الذاكرة فى جهاز الـــ PLC إلى نوعين:

-ذاكرة موجودة داخل الـــ PLC و دائماً تكون من النوع المتطاير volatile memory.

-ذاكرة تضاف إلى الـــ PLC و دائماً تكون من النوع الغير المتطاير non volatile memory.

-النوع الأول.

ذاكرة متطايرة.......RAM (Random Access Memory)

مميزات الذاكرة المتطايرة

يمكن رسم, تعديل و مسح البرنامج أكثر من مرة.

عيوب الذاكرة المتطايرة

فى حالة أنقطاع التيار يمسح البرنامج بالكامل (إذا كان بدون بطارية).

ملاحظة:

- لحل مشكلة إنقطاع التيار يتم توصيل بطارية.

-النوع الثانى.

ذاكرة غير متطايرة........ROM (Read Only Memory):

مميزات الذاكرة غير المتطايرة

فى حالة إنقطاع التيار لا يمح البرنامج (حتى إذا كان بدون بطارية).

عيوب الذاكرة غير المتطايرة

لا يمكن تعديل أو مسح البرنامج مرة أخرى بعد تنفيذه.

ملاحظة:

- توجد أنواع أخرى من مشتقات الـــ ROM وهى:

EPROM - **EEPROM** - **FLASH MEMORY**

-النوع الأول.

ذاكرة غير متطايرة

(Electrical Programmable Read Only Memory) EPROM

- كيفية كتابة البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الـــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, و بالضغط على "تحميل" يتم تحميل البرنامج على الذاكرة.

- كيفية مسح البرنامج:

يتم مسح البرنامج من على هذا النوع من الذاكرة عن طريق الأشعة الفوق البنفسجية بحيث يتم وضع الذاكرة داخل جهاز الأشعة الفوق البنفسجية لزمن محدد ويتم تنفيذ هذه العملية من قبل شخص ذات خبرة لتجنب تلف الذاكرة.

-النوع الثانى.

ذاكرة غير متطايرة

(Erasable Electrical Programmable Read Only Memory) EEPROM

- كيفية كتابة البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الـــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, بالضغط على "تحميل" يتم تحميل البرنامج على الذاكرة.

- كيفية مسح البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الـــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, و بالضغط على erase memory cartridge يتم مسح جميع محتويات الذاكرة.

-النوع الثالث.

ذاكرة غير متطايرة

FLASH MEMORY

- كيفية كتابة البرنامج:

يمكن بكل سهولة كتابة البرنامج على هذا النوع من الذاكرة دون أى مشكلة و دون استخدام كرت البرمجة أى التحكم بالذاكرة بدون فصلها عن جهاز الـــ PLC.

- كيفية مسح البرنامج:

يمكن بكل سهولة مسح البرنامج من على هذا النوع من الذاكرة دون أى مشكلة و دون استخدام كرت البرمجة أى التحكم بالذاكرة بدون فصلها عن جهاز الـــ PLC.

ملاحظة:

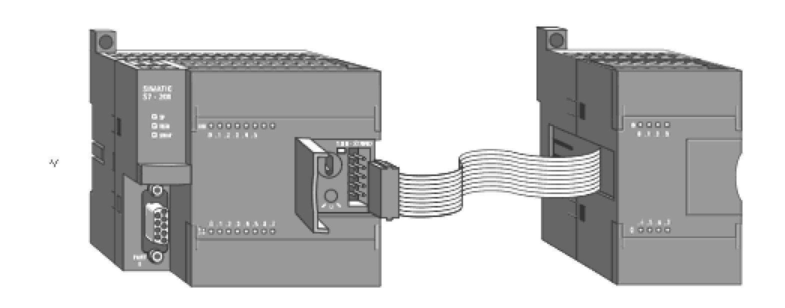
- قبل كتابة البرنامج على أى ذاكرة من نوع

EPROM أو EEPROM أو FLASH MEMORY

يجب تجربة البرنامج على الذاكرة الـــ RAM عدة مرات للتأكد من عدم وجود أى مشاكل بالبرنامج لأن كثرة المسح أو التعديل على الذاكرة الغير متطيرة يؤثر على العمر الإفتراضى للذاكرة.

٧ - وحدات دخل أو خرج إضافية.........Extension Module:

نظراً لأن فى بعض الحالات قد يحتاج المبرمج إلى مجموعة مداخل أو مخارج إضافية لاستخدامها فى البرمجة فأنة من الممكن شراء وحدات تحتوى على عدد محدد من المداخل فقط أو وحدات تحتوى على عدد محدد من المخارج فقط أو وحدات تحتوى على عدد محدد من مداخل و المخارج معاً.



مثال لوحدات الدخل و الخرج الإضافية:

-وحدات دخل و خرج رقمية إضافية:

١- EM221: وحدة إضافية تحتوى على ثمانية مداخل.

٢- EM222: وحدة إضافية تحتوى على ثمانية مخارج.

٣ - EM223: وحدة إضافية تحتوى على أربعة مداخل و أربعة مخارج.

٤ - EM223: وحدة إضافية تحتوى على ثمانية مداخل و ثمانية مخارج.

٥- EM223: وحدة إضافية تحتوى على ستة عشر دخل و ستة عشر خرج.

-وحدات دخل و خرج تناظرية إضافية:

١- EM231: وحدة إضافية تحتوى على أربعة مداخل.

٢- EM232: وحدة إضافية تحتوى على مخرجان.

٣ - EM235: وحدة إضافية تحتوى على أربعة مداخل و خرج واحد.

ملاحظة:

يختلف توصيل الوحدات الإضافية بين بعض أجهزة الـــ PLCS, فمثلاً:

- فى حالة توصيل وحدة أضافية لجهاز PLC CPU 224 يتم توصيل الوحدة الإضافية إلى جهاز الـــ PLC عن طريق كابل.
- فى حالة توصيل وحدة إضافية لجهاز PLC CPU 214 يتم توصيل الوحدة الإضافية إلى جهاز الـــ PLC عن طريق تركيب الاثنان معاً أى بالتوصيل المباشر.

يفضل توصيل الوحدات الإضافية إلى وحدة الـــ PLC بواسطة الكابل وهذا لأنه يمكن وضع الاثنين تحت بعضهم أو على مسافة بعيدة.

أجهزة للتحكم فى وحدة الـــ PLC:

من الممكن التحكم فى جهاز الـــ PLC عن طريق:

Computer - HMI

أى أنه يمكن التحكم بوحدة التحكم المنطقى باستخدام شاشة التحكم أو باستخدام الكمبيوتر ويمكن أيضاً استخدام الاثنين معاً

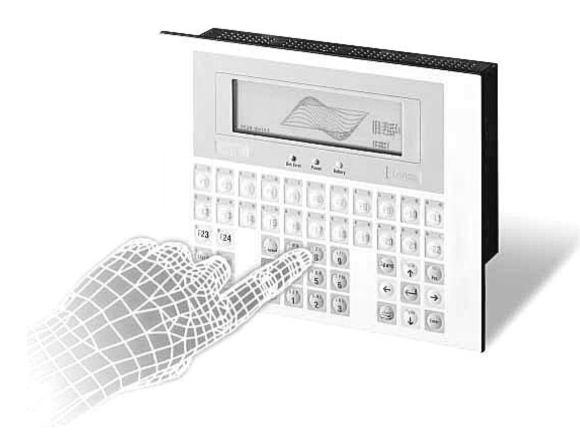
١– الكمبيوتر........Computer:

حيث يتم بسهولة عمل أى برنامج بأى لغة برمجة, تحميل البرنامج إلى جهاز الـــ PLC, أخذ البرنامج من جهاز الـــ PLC, التعديل فى البرنامج بالأضافة أو بالمسح, معرفة حالة البرنامج و كيف يعمل و تغير بعض البيانات دون أقاف البرنامج....الخ



٢– شاشة التحكم........(Human Machine Interface):

حيث يتم بسهولة التعديل فى البرنامج فى حدود معينة, معرفة حالة البرنامج و كيف يعمل, تغير بعض البيانات دون أقاف البرنامج, رؤية الألرمات على الشاشة و معرفة نوع العطل, أطفاء الألرمات....الخ



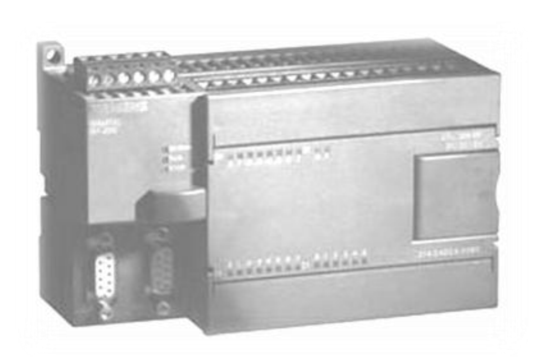
ملاحظة:

١- يستخدم جهاز الكمبيوتر فى البداية فقط لعمل البرنامج بالكامل ويفضل بعد ذالك أستبدال الكمبيوتر بشاشة التحكم نظراً لصغر حجمها.

٢ - بعض أجهزة الـــ PLC تحتوى على مخرج واحد للبرمجة مثل جهاز PLC (CPU 224) والبعض الأخر يحتوى على مخرجين للبرمجة مثل جهاز PLC (CPU 224 XP) و لذالك فى الـــ CPU 224 يتم توصيل أما جهاز كمبيوتر أما شاشة تحكم بينما فى الـــ CPU 224 XP من الممكن توصيل جهاز كمبيوتر و شاشة تحكم معاً.

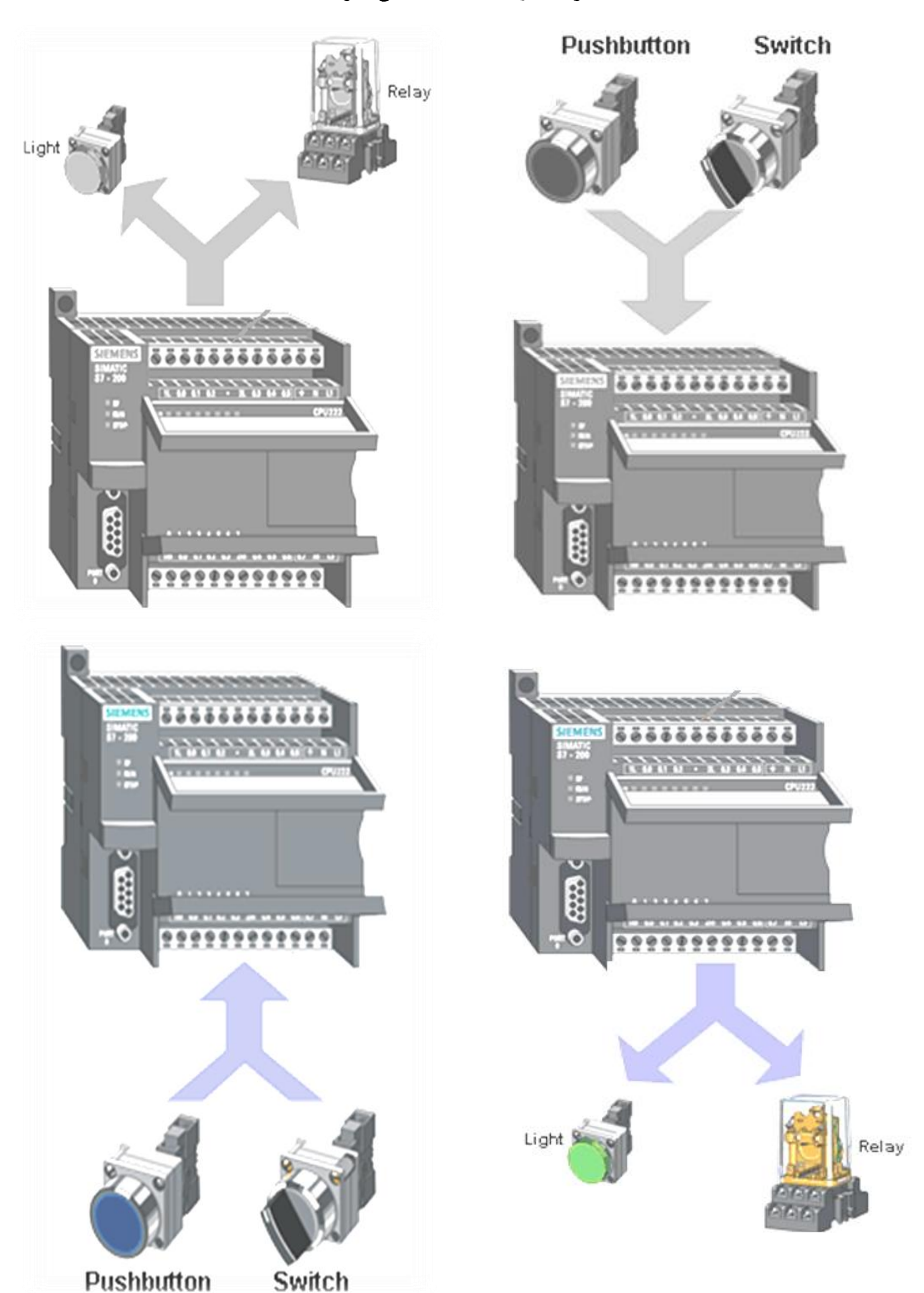


CPU 224



CPU 224 XP

ملاحظة هامة: يختلف مكان المدخلات و المخرجات بين بعض الموديلات.



الباب الثانى

الذاكرة

- تمهيـــــــد عن الذاكرة داخل جهاز الــ PLC.
- أحجام الذاكـــــرة المستخدمة فى جهاز الــ PLC.
- كيفية كتابة بيانات داخل الذاكـــــــــــرة.
- كيفية قراءة بيانات من الذاكـــــــــــرة.
- نظـــــــــــــــام الأعداد الثنائية.
- نظـــــــــــــــام الأعداد العشرية.
- نظـــــــــــــــام الأعداد السداسية عشر.
- نظـــــــــــــــام الأعداد العشرية المكودة ثنائياً.
- نظـــــــــــــــام العلامة العشرية.
- العمليات الحسابية للأعداد الثنائيـــــــــــة.
- كـــيـــفـــيـــة التحـــويـــل من نظـــــام إلى أخر.

أحجم الذاكرة المستخدمة فى جهاز الـــ PLC :

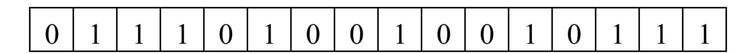
١- BIT: هى أصغر وحدة للذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر أو واحد.

1

٢- BYTE: هى ذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 8 bits.

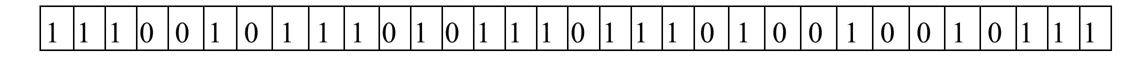
1	0	1	0	0	0	1	1

٣- WORD: هى ذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 2 bytes أو 16 bits.



0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

٤- D.WORD: هى أكبر وحدة للذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر و واحد وهو تتكون من 2 words أو 4 bytes أو 32 bits.



1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

هام:

لكل وحدة من الذاكرة أسم و طريقة للكتابة:

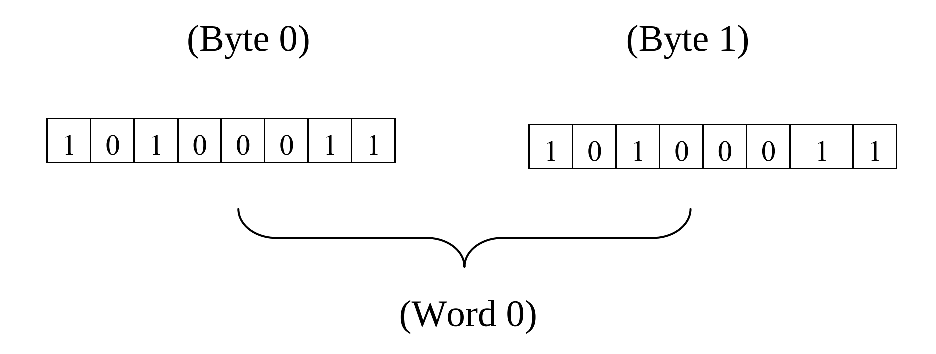
- BIT 0, BIT 1, BIT 2, BIT 3, BIT 4, ...
- BYTE 0, BYTE 1, BYTE 2, BYTE 3, ...
- WORD 0, WORD 2, WORD 4, WORD 6, ...
- D.WORD 0, D.WORD 4, D.WORD 8, D.WORD 12

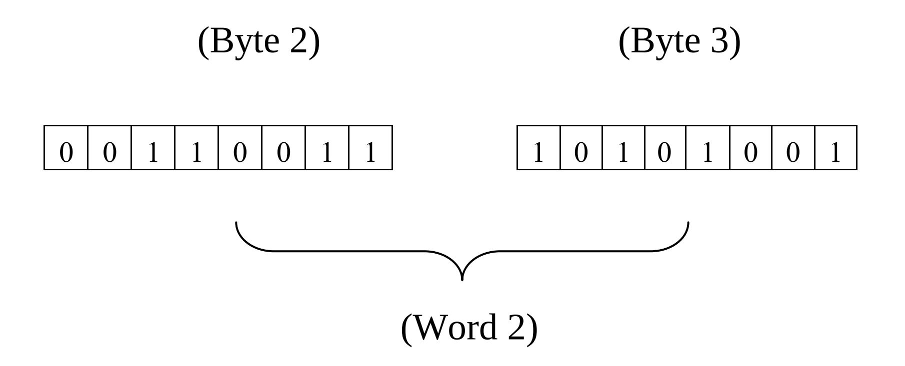
ملاحظة:

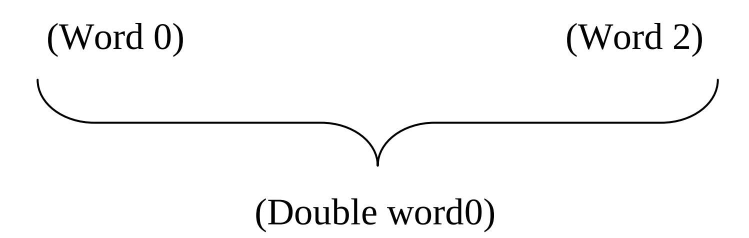
من المهم جداً مراعاة أن فى حالة كتابة معلومات بواسطة الـــ bit يجب تحديد أسم الـــ bit.

فى حالة كتابة معلومات بواسطة الـــ byte أو الـــ word أو الـــ Dword يجب بدء الكتابة من اليمين إلى اليسار وهذا لأن جميع أحجام الذاكرة هى مكونة من مجموعة من الـــ bits وتستخدم نفس الطريقة أيضاً فى حالة قراءة البيانات من على الذاكرة.

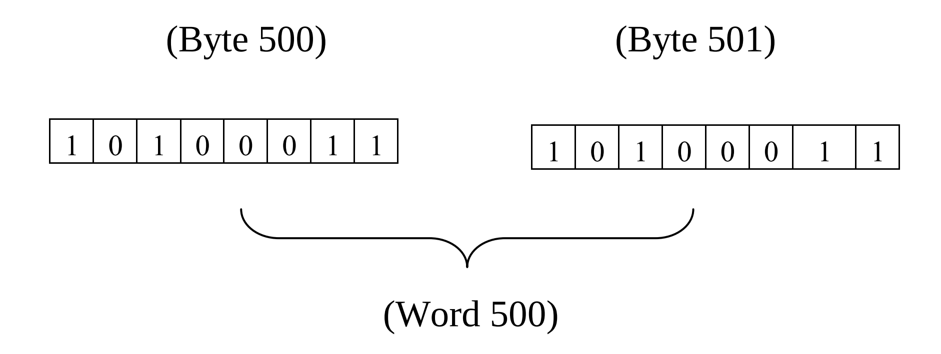
شرح مفصل:

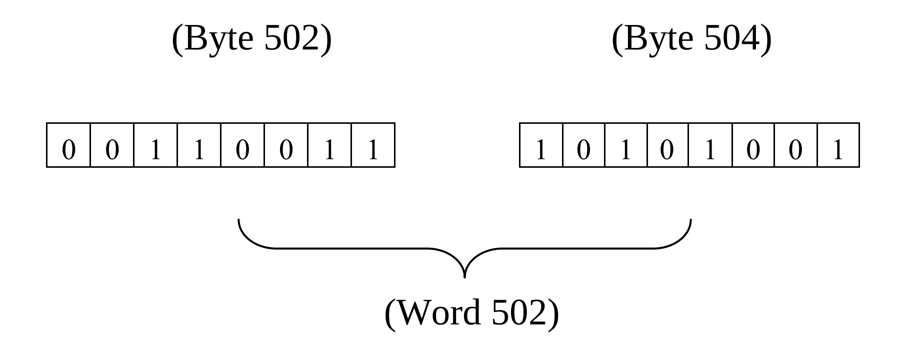


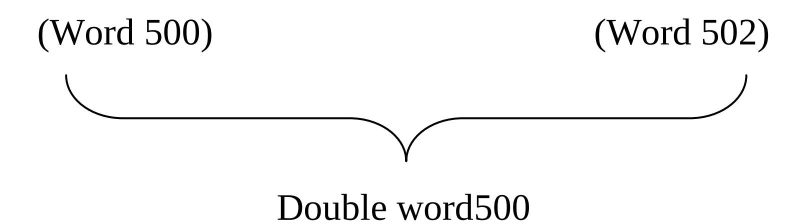



(Word 0)
(Word 2)
(Double word0)

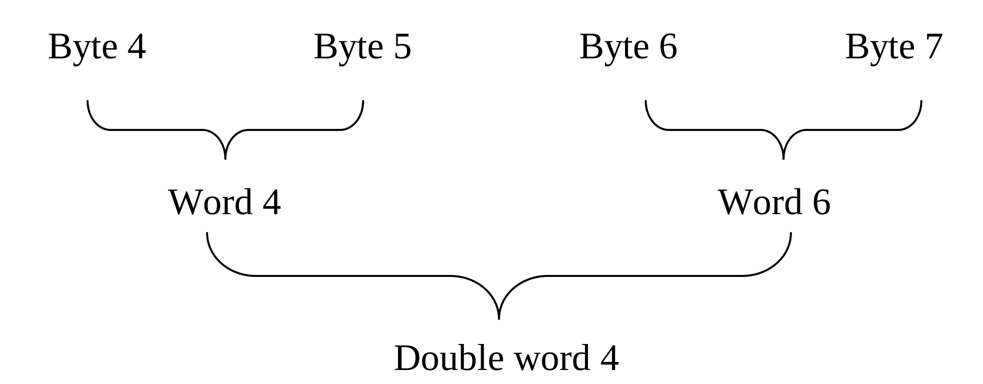
مثال أخر:


(Byte 500)
(Byte 501)
1 0 1 0 0 0 1 1
1 0 1 0 0 0 1 1
(Word 500)

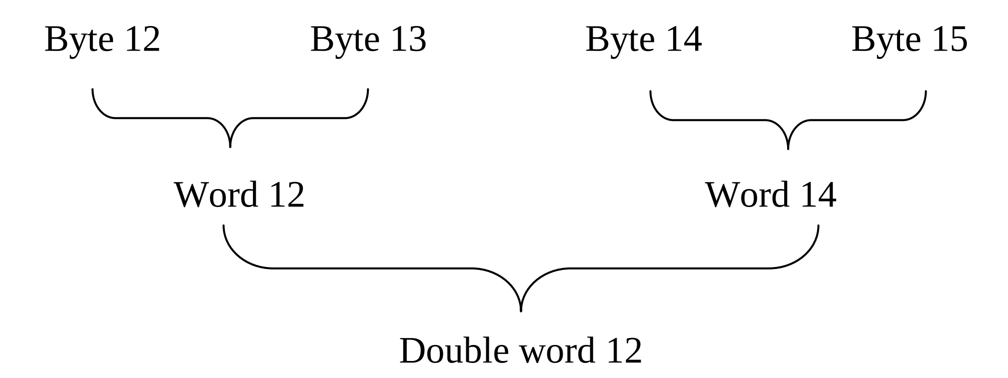

(Byte 502)
(Byte 504)
0 0 1 1 0 0 1 1
1 0 1 0 1 0 0 1
(Word 502)



مثال أخر:



مثال أخر:



تكوين الـــ byte من الداخل:

bit0.7 bit0.6 bit0.5 bit0.4 bit0.3 bit0.2 bit0.1 bit0.0

Byte 0

مثال أخر

bit11.7 bit11.6 bit11.5 bit11.4 bit11.3 bit11.2 bit11.1 bit11.0

Byte 11

كيفية قراءة ما بداخل الذاكرة (حسب المثال المذكور فى صفحة 61):

- فى حالة قراءة أى bit تتم القراءة من اليمين إلى اليسار ولكن لذكر أسم الـــ bit يجب تحديد لأى byte ينتمى هذا الـــ bit فمثلاً: b2.0 هو الـــ bit الأول فى الـــ byte الثالث أو الـــ b5.6 هو الـــ bit السابع فى الـــ byte السادس.

- فى حالة قراءة أى byte من الداخل تتم القراءة من اليمين إلى اليسار لأن الـــ byte يتكون من مجموعة bits وبعد قراءة ذالك الـــ byte تتم قراءة الـــ byte الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة bytes فمثلاً يقرأ الـــ byte 0 ثم يقرأ الـــ byte الذى على يمينة الذى هو byte 1 وهكذا.

- فى حالة قراءة أى word من الداخل تتم القراءة من اليمين إلى اليسار لأن الــ word يتكون من مجموعة bits وبعد قراءة ذالك الــ word تتم قراءة الــ word الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة words فمثلاً يقرأ الــ word 0 ثم يقرأ الــ word الذى على يمينه الذى هو word 2 وهكذا وبالنسبة إلى الــ word 2 يقرأ الــ bits من اليمين إلى اليسار أى يقرأ الــ byte 3 ثم الــ byte2.

- فى حالة قراءة أى D.word من الداخل تتم القراءة من اليمين إلى اليسار لأن الــ D.word يتكون من مجموعة bits وبعد قراءة ذالك الــ D.word تتم قراءة الــ D.word الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة D.words فمثلاً يقرأ D.word0 ثم يقرأ الــ D.word الذى على يمينه الذى هو D.word4 وهكذا.

بالنسبة إلى الــ D.word0 يقرأ الــ bits من اليمين إلى اليسار أى يقرأ الــ byte3 ثم الــ byte2 ثم الــ byte1 ثم الــ byte0 أى أنه يقرأ الــ word2 ثم الــ word0.

النظم الرقمية المستخدمة فى جهاز الـــ PLC :

١- نظام الأعداد الثنائية (binary).

٢- نظام الأعداد العشرية (decimal).

٣ - نظام الأعداد السداسية عشر (hexadecimal).

٤ - نظام الأعداد العشرية المكودة ثنائياً (BCD).

٥ - نظام العلامة العشرية (real).

٦ - نظام العلامة العشرية (floating point).

٧ - نظام أسكى (ASCII).

١– النظام الثنائى (**binary**).

يتكون من:

(صفر) و (١).

مثال:

$(10010111)_2$.

لذلك لكتابة أرقام من نوع النظام الثنائى binary يمكن استخدام ذاكرة بحجم:

- BIT.
- BYTE.
- WORD.
- DOUBLE WORD.

٢– النظام العشرى (decimal).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

$(55)_{10}$.

لذلك لكتابة أرقام من نوع النظام العشرى decimal **يمكن استخدام ذاكرة بحجم:**

- BYTE.
- WORD.
- DOUBLE WORD.

٣ - النظام السداسى عشر (hexadecimal).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) , (٩), (A), (B), (C), (D), (E) و (F).

مثال:

$(6A2F)_{16}$.

لذلك لكتابة أرقام من نوع النظام السداسي عشر hexadecimal **يمكن أستخدام ذاكرة بحجم:**

- BYTE.
- WORD.
- DOUBLE WORD.

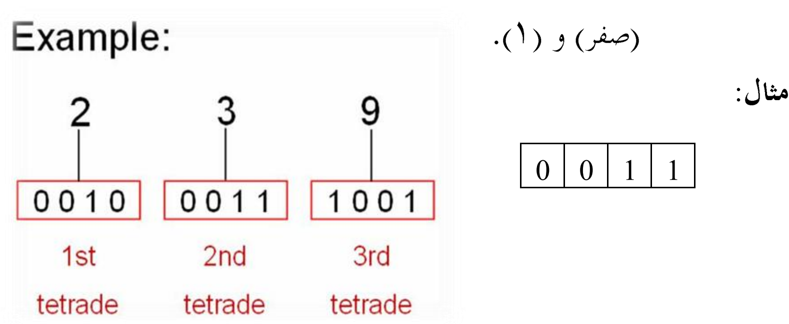
٤ - نظام الأعداد العشرية المكودة ثنائياً (Binary Code Decimal).



يتكون من:

(صفر) و (١).

مثال:



لذلك لكتابة أرقام من نوع نظام الـــ BCD يمكن أستخدام ذاكرة بحجم:

- BYTE.
- WORD.
- DOUBLE WORD.

٥ - نظام العلامة العشرية (real).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

52.14

لكتابة أرقام من نوع نظام العلامة REAL يمكن أستخدام فقط ذاكرة بحجم:

- DOUBLE WORD.

٦ - نظام العلامة العشرية (floating point).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

+5.2120000 + E 1

لكتابة أرقام من نوع نظام العلامة FLOATING **يمكن أستخدام فقط ذاكرة بحجم:**

- DOUBLE WORD.

٧- نظام أسكى (American Standard Code for Information Interchange).

هو جدول يحتوى على جميع الأرقام و الحروف و الرموز التى يمكن كتابتها بواسطة لوحة المفاتيح كما هو موضح بالصورة, لأن الـــ PLC لا يفهم سوى الصفر و الواحد.

الجدول:

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

النظم الرقمية هامه جداً لأنه من الممكن قراءة أو كتابة البيانات بنظم رقمية مختلفة, فمثلاً:

- فى حالة التعامل مع دخل أو خرج من الأسهل أن تكون الصيغة format من نوع binary بحيث يكون من السهل تمييز حالة المفتاح أو الخرج, فمثلاً بالنسبة لحالة المفتاح أذا كان يساوى واحد فهذا يعنى أن المفتاح مغلق أما أذا كان يساوى صفر فهذا يعنى أن المفتاح مفتوح وهكذا أيضاً فى حالة الخرج و فى حالة الريليه الداخلى.

- فى حالة التعامل مع مؤقتات زمنية أو عدادات من الأسهل أن تكون الصيغة format من نوع decimal بحيث يكون من السهل قراءة أو كتابة أى زمن للمؤقت أو أى عدد للعداد.

التحويل من نظم رقمية إلى نظم رقمية أخرى مستخدمة فى جهاز الــ PLC :

- **من الــ decimal إلى الــ binary:**

يتم التحويل من الــ decimal إلى الــ binary بواسطة القسمة على أثنين, فمثلاً:
رقم 100:

100 / 2 = 50	الباقى ⇨	0
50 / 2 = 25	الباقى ⇨	0
25 / 2 = 12	الباقى ⇨	1
12 / 2 = 6	الباقى ⇨	0
6 / 2 = 3	الباقى ⇨	0
3 / 2 = 1	الباقى ⇨	1
1	الباقى ⇨	1

⇨ $(100)_{10} = (1100100)_2$

يكتب المثال السابق بهذه الطريقة للتبسيط:

100	2	0	
50	2	0	
25	2	1	
12	2	0	**الباقي**
6	2	0	
3	2	1	
1	2	1	

تتم القسمة على أثنان لأن الأساس الخاص بنظام الأعداد الثنائية هو أثنان.

مثال أخر:

رقم $(255)_{10}$

255 / 2 = 127 الباقي ⇨ 1
127 / 2 = 63 الباقي ⇨ 1
63 / 2 = 31 الباقي ⇨ 1
31 / 2 = 15 الباقي ⇨ 1
15 / 2 = 7 الباقي ⇨ 1
7 / 2 = 3 الباقي ⇨ 1
3 / 2 = 1 الباقي ⇨ 1
1 ⇨ 1

⇨ $(255)_{10} = (11111111)_2$

يكتب المثال السابق بهذه الطريقة للتبسيط:

255	2	1	
127	2	1	
63	2	1	
31	2	1	**الناتج**
15	2	1	
7	2	1	
3	2	1	
1	2	1	

بعض الأمثلة على التحويل من الــ decimal **إلى الــ** binary.

Binary numbers						
DECIAML		BINARY	DECIMAL		BINNARY
١	→	000000001	٥١	→	00110011
٢	→	00000010	٥٢	→	00110100
٣	→	00000011	٥٣	→	00110101
٤	→	00000100	٥٤	→	00110110
٥	→	00000101	٥٥	→	00110111
٦	→	00000110	٥٦	→	00111000
٧	→	00000111	٥٧	→	00111001
٨	→	00001000	٥٨	→	00111010
٩	→	00001001	٥٩	→	00111011
١٠	→	00001010	٦٠	→	00111100
١١	→	00001011	٦١	→	00111101
١٢	→	00001100	٦٢	→	00111110
١٣	→	00001101	٦٣	→	00111111
١٤	→	00001110	٦٤	→	01000000
١٥	→	00001111	٦٥	→	01000001
١٦	→	00010000	٦٦	→	01000010
١٧	→	00010001	٦٧	→	01000011
١٨	→	00010010	٦٨	→	01000100
إلخ		إلخ	إلخ		إلخ

- **من الــ binary إلى الــ decimal**:

لكل bit قيمة ثابتة, فمثلاً الــ bit0 قيمته تساوى 2^0 أى يساوى 1 أذا قمنا بنفس الشئ فى جميع الــ bits فالقيم سوف تكون:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الــ byte هو 255 فى هذه الحالة.							

يتم التحويل من الــ binary إلى الــ decimal بواسطة ضرب محتوى الــ bit فى القيمة الخاصة به, فمثلاً:

١ - رقم $(11011100)_2$
٢ - رقم $(11001110)_2$
٣ - رقم $(01001111)_2$

المثال الأول:

رقم $(11011100)_2$

1	**1**	**0**	**1**	**1**	**1**	**0**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	0	16	8	4	0	0
المجموع الكلى لمحتوى الــ byte هو 220 فى هذه الحالة.							

المثال الثانى:

رقم $(11001110)_2$

0	**0**	**0**	**0**	**1**	**1**	**1**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	0	0	8	4	2	0
المجموع الكلى لمحتوى الـــ byte هو 206 فى هذه الحالة.							

المثال الثالث:

رقم $(01001111)_2$

0	**1**	**0**	**0**	**1**	**1**	**1**	**1**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	64	0	0	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 79 فى هذه الحالة.							

- من الـــ **decimal** إلى الـــ **hexadecimal**:

يتم التحويل من الـــ decimal إلى الـــ hexadecimal بواسطة القسمة على 16, فمثلاً:

١- رقم: 100

٢- رقم: 50

المثال الأول:

رقم 100

100 / 16 = 6 الباقى ⇨ 4

6 = 6 الباقى ⇨ 6

⇨ $(100)_{10} = (64)_{16}$

يكتب المثال السابق بهذه الطريقة للتبسيط:

100	16	4	
6		6	**الناتج**

المثال الثانى:

رقم 50

50 / 16 = 3 الباقى ⇨ 2

3 = 3 الباقى ⇨ 3

⇨ $(50)_{10} = (32)_{16}$

يكتب المثال السابق بهذه الطريقة للتبسيط:

50	16	2	
3		3	**الناتج**

- **من الـ hexadecimal إلى الـ decimal**:

يتم التحويل من الـ hexadecimal إلى الـ decimal بواسطة الضرب فى 16, فمثلاً:

١- رقم 101A:

٢- رقم A0F:

المثال الأول:

رقم 101A

1	**0**	**1**	**A**
16^3	16^2	16^1	16^0
4096	0	16	10
4122			

المثال الثانى:

رقم 200F

2	**0**	**0**	**F**
16^3	16^2	16^1	16^0
8192	0	0	15
8207			

- **من الـــ hexadecimal إلى الـــ binary**:

يتم تحويل أى رقم hexadecimal إلى أربع أرقام binary, فمثلاً:

١- رقم 1A:

٢- رقم AF:

المثال الأول:

رقم 1A

1				**A**			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
0	0	0	1	1	0	1	0
المجموع الكلى لمحتوى الـــ byte هو: **00011010**							

المثال الثانى:

رقم AF

A				**F**			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
1	0	1	0	1	1	1	1
المجموع الكلى لمحتوى الـــ byte هو: **10101111**							

- **من الـــ binary إلى الـــ hexadecimal**:

يتم تحويل كل أربع أرقام رقم من الـــ binary إلى ما يعادلها في الـــ decimal ثم تكتب بصيغة الـــ hexadecimal.

١- رقم $(11001010)_2$:
٢- رقم $(00010111)_2$:

المثال الأول:

رقم $(11001010)_2$

1	1	0	0	1	0	1	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
12				10			
تصبح القيمة بلغة الـــ hexadecimal: **$(CA)_{16}$**							

المثال الثاني:

رقم $(00010111)_2$

0	0	0	1	0	1	1	1
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
1				7			
تصبح القيمة بلغة الـــ hexadecimal: **$(17)_{16}$**							

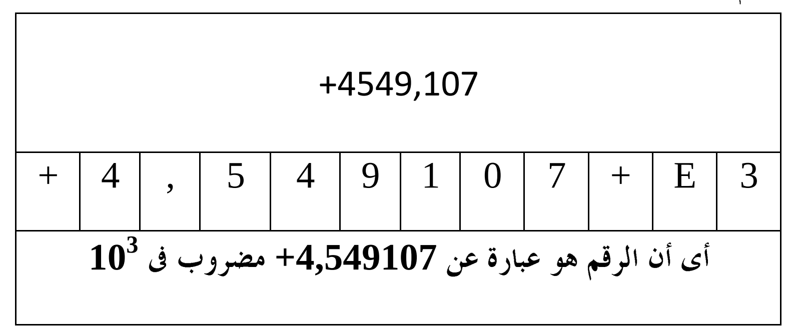
- **من الــ real إلى الــ floating point**:

تتم كتابة الإشارة لتوضيح أذا كان الرقم موجب أم سالب و بعد ذالك يكتب الرقم الأول ثم العلامة العشرية ثم تتم كتابة باقى الأرقم وبعد ذالك يكتب الأس لكى يعادل الرقم المكتوب بالــ real.

١- رقم 4549,107+
٢- رقم 719,17-

المثال الأول:

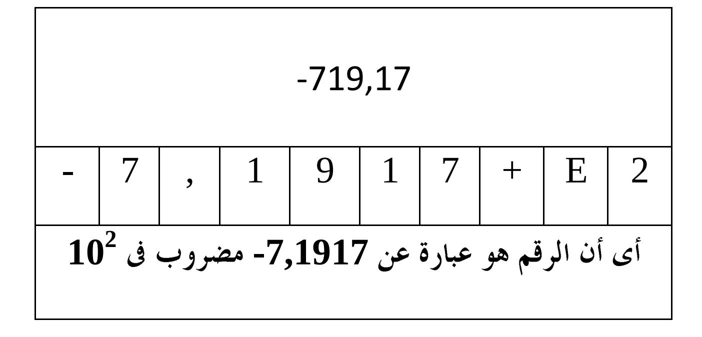
رقم 4549,107+



+4549,107											
+	4	,	5	4	9	1	0	7	+	E	3
أى أن الرقم هو عبارة عن 4,549107+ مضروب فى 10^3											

المثال الثانى:

رقم 719,17-



-719,17										
-	7	,	1	9	1	7	+	E	2	
أى أن الرقم هو عبارة عن 7,1917- مضروب فى 10^2										

- **من الـــ floating point إلى الـــ real**:

تتم كتابة الإشارة ثم الرقم ثم وضع العلامة العشرية حسب الأس ثم تتم كتابة باقى الرقم لكى يعادل الرقم المكتوب بالـــ floating point.

١- رقم +4,549107+E3
٢- رقم -6,11107+E3

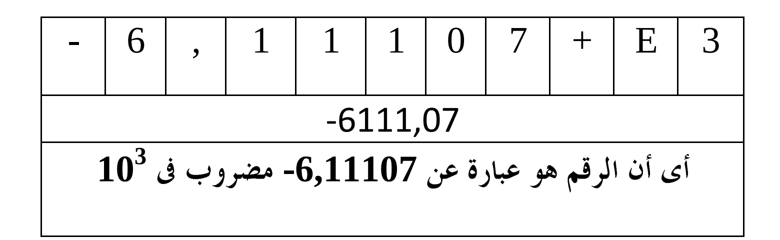
المثال الأول:

رقم +4,549107+E3

+	4	,	5	4	9	1	0	7	+	E	3
+4549,107											
أى أن الرقم هو عبارة عن 4,549107+ مضروب فى 10^3											

المثال الثانى:

رقم -6,11107+E3



-	6	,	1	1	1	0	7	+	E	3
-6111,07										
أى أن الرقم هو عبارة عن 6,11107- مضروب فى 10^3										

الأرقام الصحيحة........Integer Math :

تنقسم الأرقام الصحيحة إلى أرقام دون إشارة وأرقام بإشارة و تنقسم الأرقام بالإشارة إلى أرقام موجبة و أرقام سالبة.

- **أرقام بدون إشارة.........Unsigned:**

نظراً أنه لا توجد إشارة أذن نقوم باستخدام جميع الـــ bits.

فمثلاً:

- **أكبر رقم صحيح يمكن كتابته على الـــ byte هو:**

1	1	1	1	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 255 فى هذه الحالة.							

- **أصغر رقم صحيح يمكن كتابتة على الـــ byte هو:**

0	0	0	0	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الـــ byte هو صفر فى هذه الحالة.							

- **رقم 22 باستخدام ذاكرة byte بدون إشارة:**

0	0	0	1	0	1	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	16	0	4	2	0
المجموع الكلى لمحتوى الـــ byte هو 22 فى هذه الحالة.							

- **الأرقام ذات إشارة........Signed:**

نظراً أنه توجد إشارة أذن نقوم باستخدام جميع الـــ bits للأرقام ماعدا الـــ bit الأخير على اليسار لأنه يستخدم للإشارة.

تنقسم الأرقام ذات الإشارة إلى أرقام موجبة و أرقام سالبة:

أولاً. الأرقام الموجبة:

يرمز للإشارة الموجبة برقم صفر ويكتب فى الـــ bit الأخير على اليسار.

فمثلاً:

- **أكبر رقم موجب صحيح يمكن كتابته على الـــ byte هو:**

0	1	1	1	1	1	1	1
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 127+ فى هذه الحالة.							

- **أصغر رقم موجب صحيح يمكن كتابته على الـــ byte هو:**

0	0	0	0	0	0	0	0
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الـــ byte هو 0+ فى هذه الحالة.							

- **رقم 22+ باستخدام ذاكرة byte بدون إشارة:**

0	0	0	1	0	1	1	0
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	0	0	16	0	4	2	0
المجموع الكلى لمحتوى الـــ byte هو 22+ فى هذه الحالة.							

ثانياً. الأرقام السالبة:

يرمز للإشارة السالبة برقم واحد ويكتب فى الـــ bit الأخير على اليسار.

فمثلاً:

- **أكبر رقم سالب صحيح يمكن كتابته على الــ byte هو:**

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الــ byte هو 1- فى هذه الحالة.							

- **أصغر رقم سالب صحيح يمكن كتابته على الــ byte هو:**

1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الــ byte هو 128- فى هذه الحالة.							

- **رقم 22– بأستخدام ذاكرة byte بدون إشارة:**

1	**1**	**1**	**0**	**1**	**0**	**1**	**0**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	0	8	0	2	0
المجموع الكلى لمحتوى الــ byte هو 22- فى هذه الحالة.							

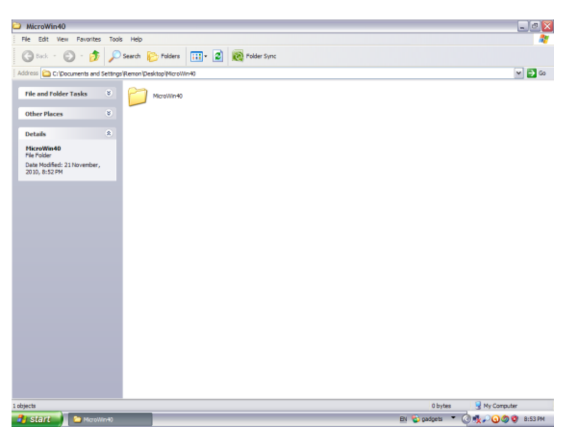
الباب الثالث

البرنامج

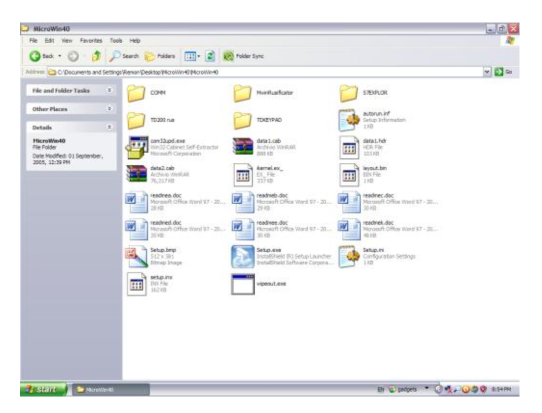
- كيفية تثبيت البرنامج.
- تعريف جهاز الـ PLC على الـ computer.
- تحديد المتغيرات الخاصة بالـ communication.
- تحديد المتغيرات الخاصة بالـ PG/PC interface.
- تحديد المتغيرات الخاصة بالـ acccess path.
- تحديد المتغيرات الخاصة بالـ PPI.
- تحديد المتغيرات الخاصة بالـ local connection.
- كيفية كشف الأخطاء قبل التشغيل.
- كيفية أيقاف جهاز الـ PLC.
- خطوات تحميل أى تمرين.
- كيفية تشغيل جهاز الـ PLC.

طريقة تثبيت البرنامج:

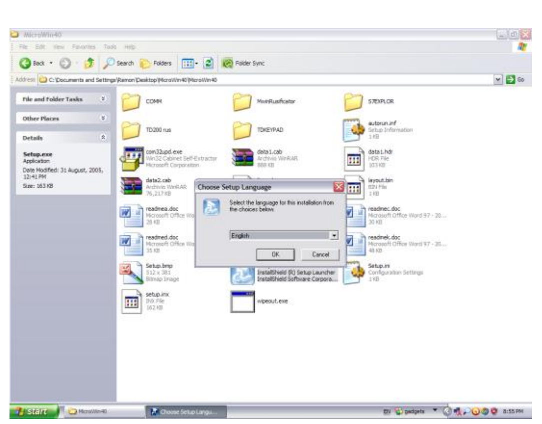
الملف الذى يحتوى البرنامج. ←



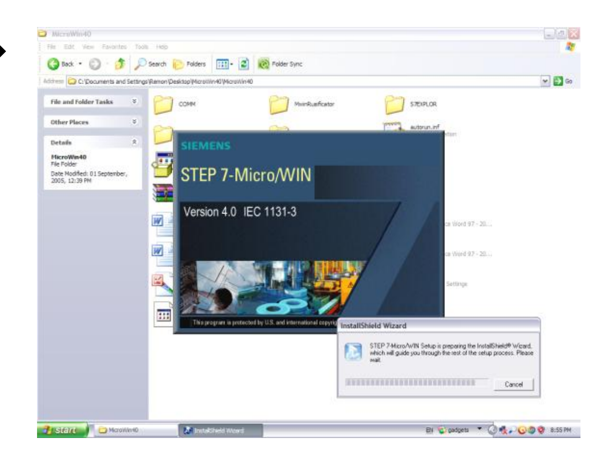
→ أضغط على ملف setup للتثبيت.



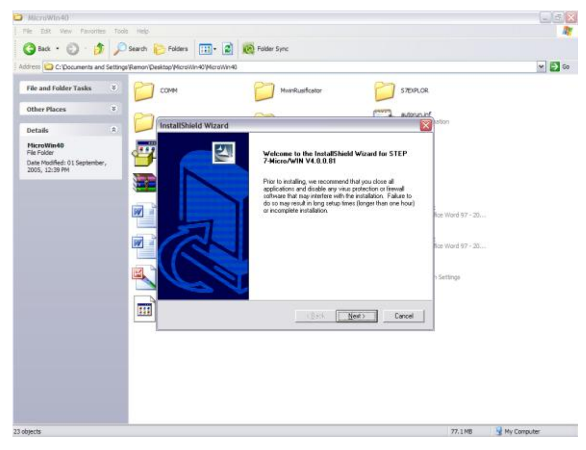
أختر اللغة المراد استخدمها. ←



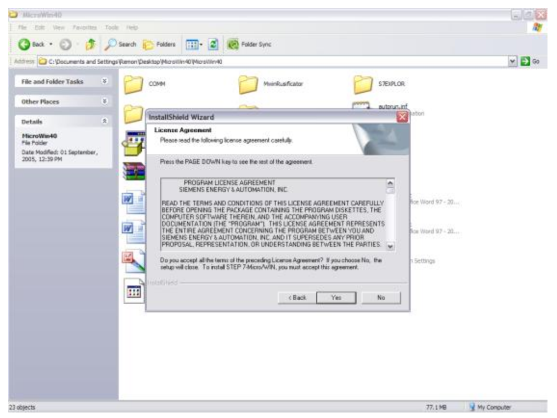
← يبدأ تثبيت البرنامج تدريجياً.



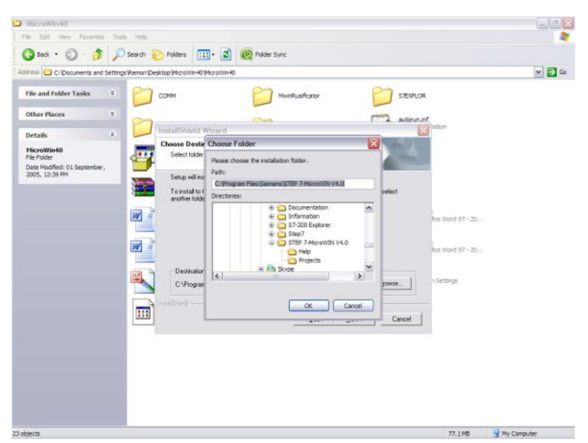
أضغط التالى next لإكمال التثبيت. →



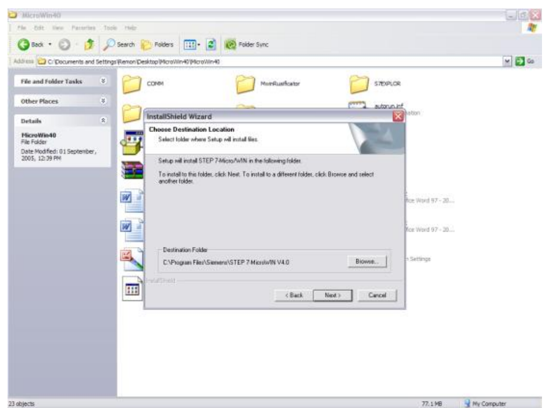
← أضغط نعم yes للموافقة على الشروط.



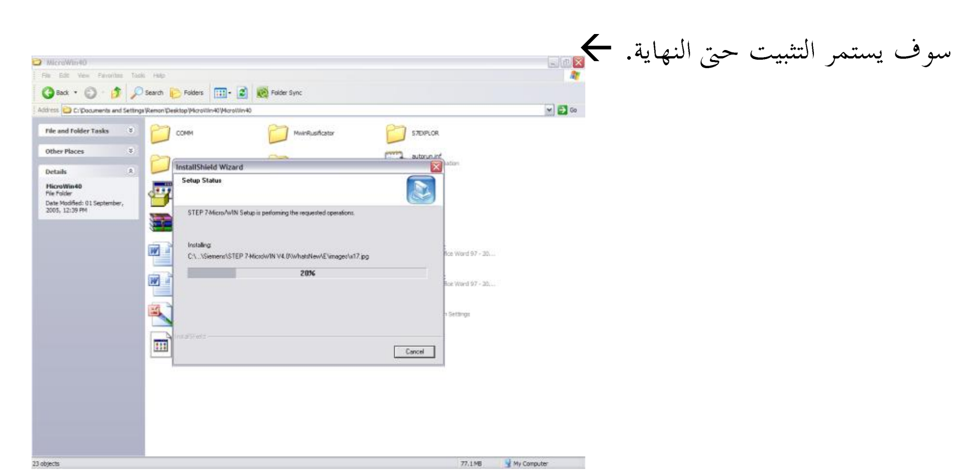
أضغط browse لاختيار مكان التثبيت. ←



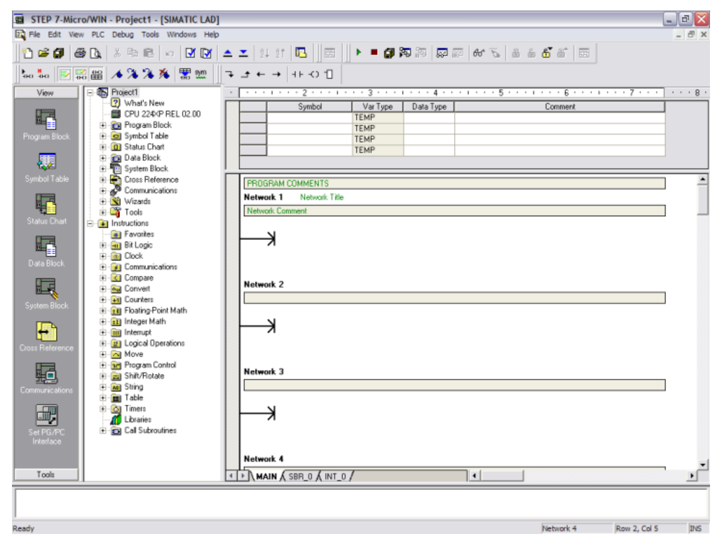
→ أضغط التالى next للأستمرار.



سوف يستمر التثبيت حتى النهاية. ←

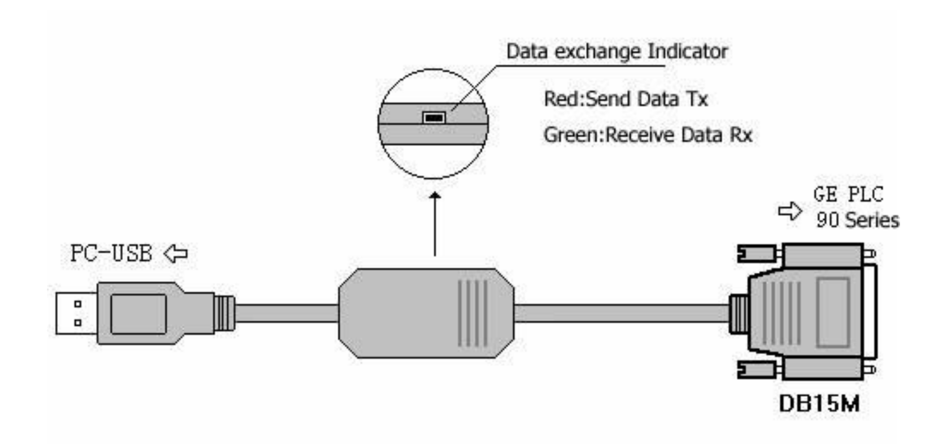


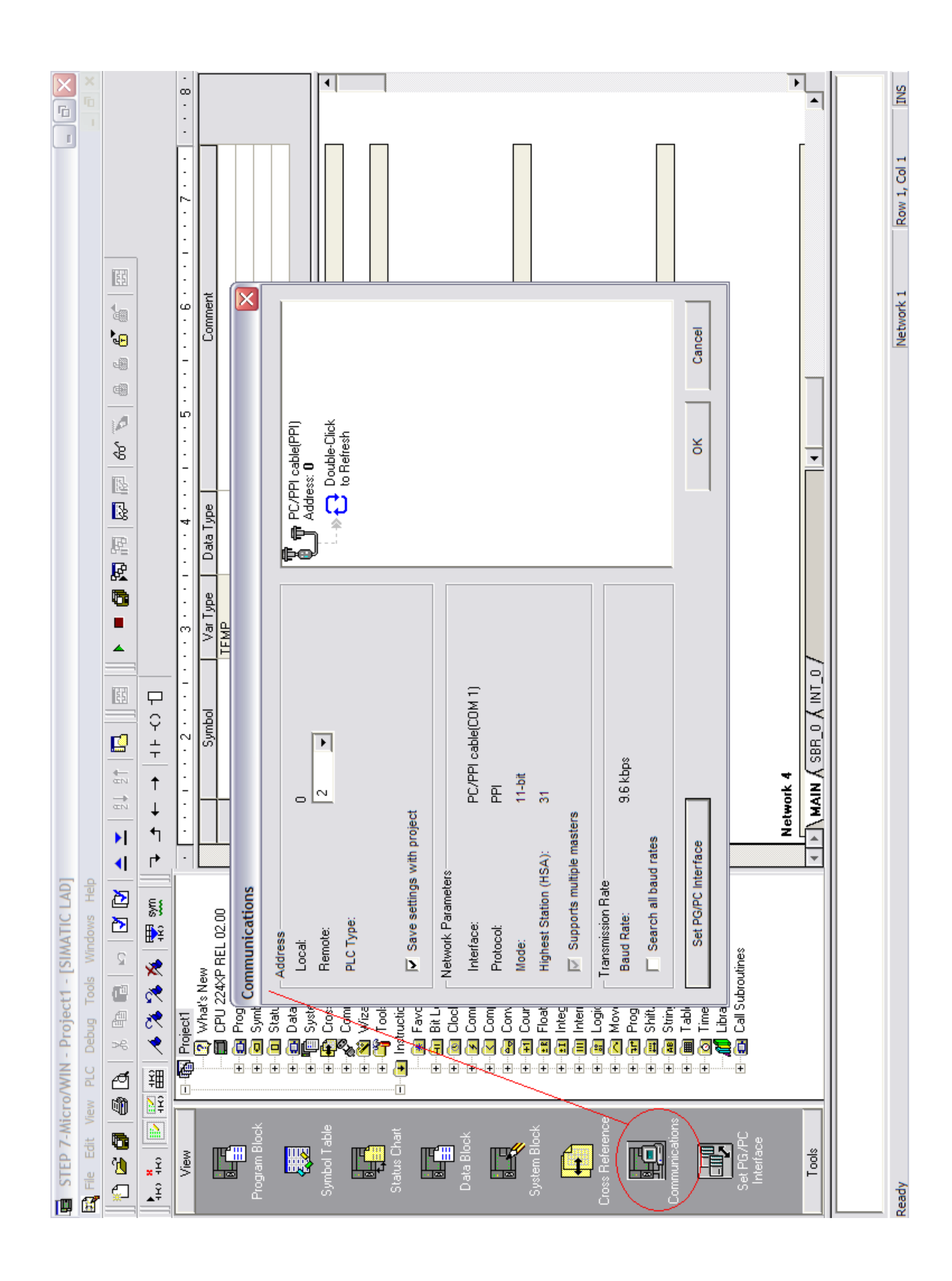
بعد تثبيت البرنامج. ←



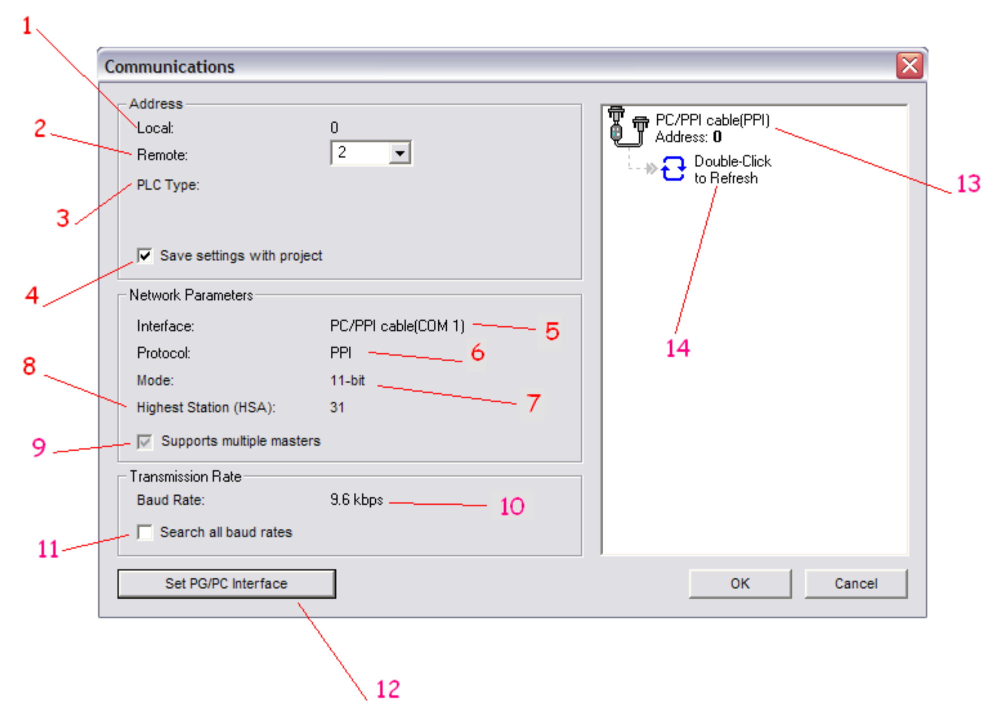
توصيل جهاز الــ PLC مع الكمبيوتر:

من المهم بعد تثبيت البرنامج الخاص بجهاز الــ PLC كما أوضح فى الخطوات السابقة أن يتم توصيل الكابل بحيث يتم توصيل الطرف الأول (Series) إلى جهاز الــ PLC و الطرف الأخر (USB) إلى الكمبيوتر وبعد ذالك يتم تعريف و تحديد بعض المتغيرات كما سوف يوضح بالصور الآتية:




STEP 7-Micro/WIN - Project1 - [SIMATIC LAD]
File Edit View PLC Debug Tools Windows Help
View
Program Block
Symbol Table
Status Chart
Data Block
System Block
Cross Reference
Communications
Set PG/PC Interface
Tools
Project1
What's New
CPU 224XP REL 02.00
Call Subroutines
Symbol
Var Type
Data Type
Comment
TEMP
Communications
Address
Local:
0
Remote:
2
PLC Type:
Save settings with project
Network Parameters
Interface:
PC/PPI cable(COM 1)
Protocol:
PPI
Mode:
11-bit
Highest Station (HSA):
31
Supports multiple masters
Transmission Rate
Baud Rate:
9.6 kbps
Search all baud rates
Set PG/PC Interface
PC/PPI cable(PPI)
Address: 0
Double-Click to Refresh
OK
Cancel
Network 4
MAIN
SBR_0
INT_0
Ready
Network 1
Row 1, Col 1
INS

صفحة الـــ COMMUNICATIONS :



١- المقصود بـــ Local هو الكمبيوتر ويرمز له برقم صفر ولكن يمكن تغير الرقم.

٢- المقصود بـــ Remote هو الــ PLC ويرمز له برقم أثنين ولكن يمكن تغير الرقم.

٣- المقصود بـــ PLC Type هو نوع الجهاز الموصل بالكابل.

٤- المقصود بـــ Save setting هو حفظ جميع المتغيرات مع البرنامج, مثلاً: لون و حجم الخط.

٥- المقصود بـــ Interface هو نوع الكابل المستخدم فالتوصيل.

٦- المقصود بـــ Protocol هو نوع النظام المتبع فالتوصيل من نقطة إلى نقطة.

٧- المقصود بـــ Mode هو نوع الـــ processor الخاص بالـــ CPU.

٨- المقصود بـــ Highest Station هو أقصى عدد من أجهزة الـــ PLC المراد التحكم بها بواسطة الكمبيوتر.

٩- المقصود بـــ Supports هو أن جهاز الــــ PLC موصل بأكثر من كمبيوتر أو HMI.

١٠- المقصود بـ Baud Rate هى سرعة نقل المعلومات من و إلى جهاز الـPLC.

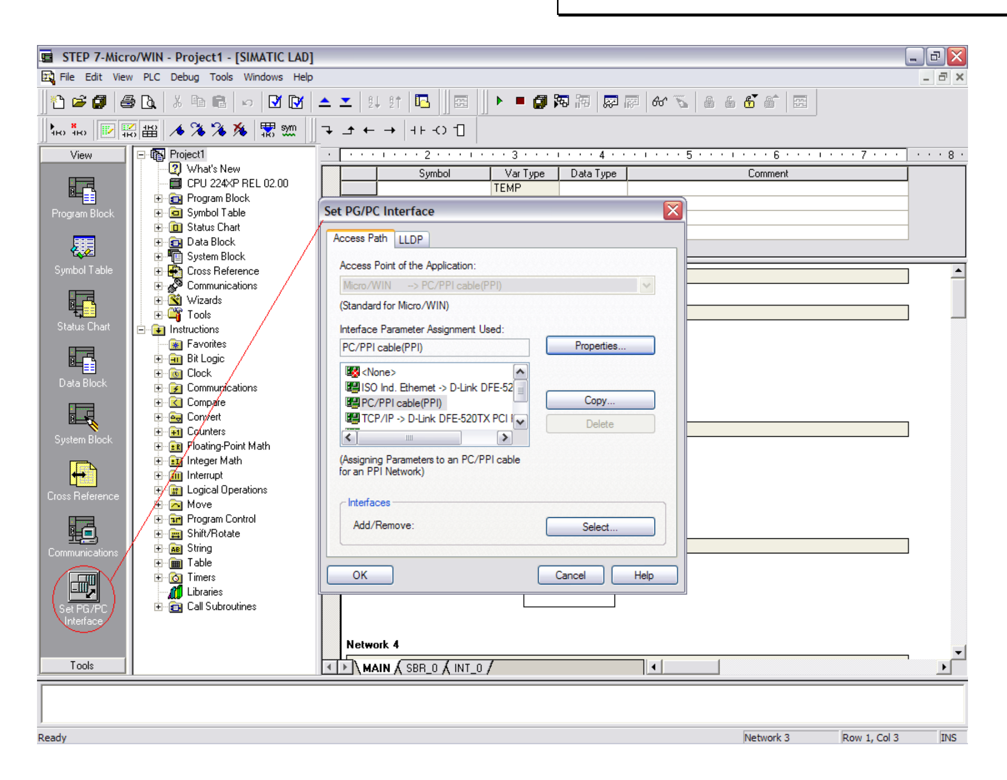
١١- المقصود بـ Search all baud rates لكى يختار سرعة نقل المعلومات من و إلى جهاز الـPLC أوتوماتيكياً.

١٢- المقصود بـ Set PG/PC هى صفحة لتحديد بعض المتغيرات الخاصة بالـ communication .

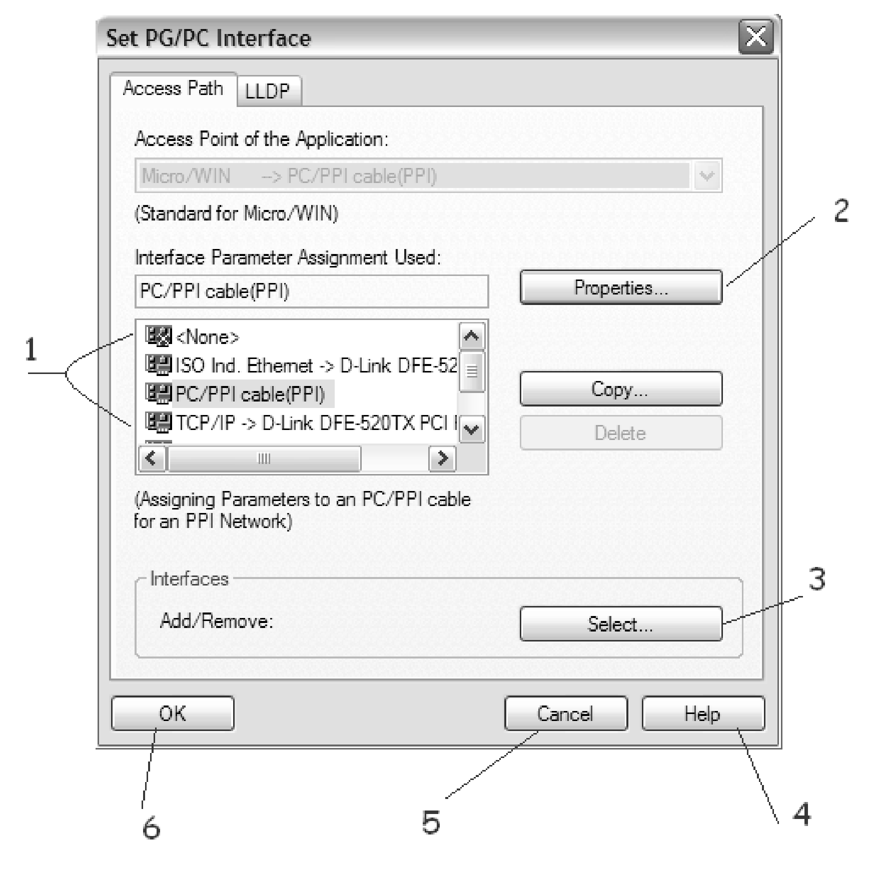
١٣- المقصود بـ PC/PPI cable يوضح رقم جهاز الـPLC و رقم الكمبيوتر فى حالة التواصل.

١٤- المقصود بـ Double click to refresh تستخدم فى حالة التوصيل لأول مرة للتوصيل.

صفحة Set PG/PC interface :



صفحة الــ ACCCESS PATH :



١- طرق مختلفة للتحكم بجهاز الــ PLC: (بواسطة الكابل أو شبكة النت أو .....).

٢ - المقصود بــ Properties هى صفحة لتحديد بعض المتغيرات الخاصة بطريقة التحكم التى سوف يتم أختيارها.

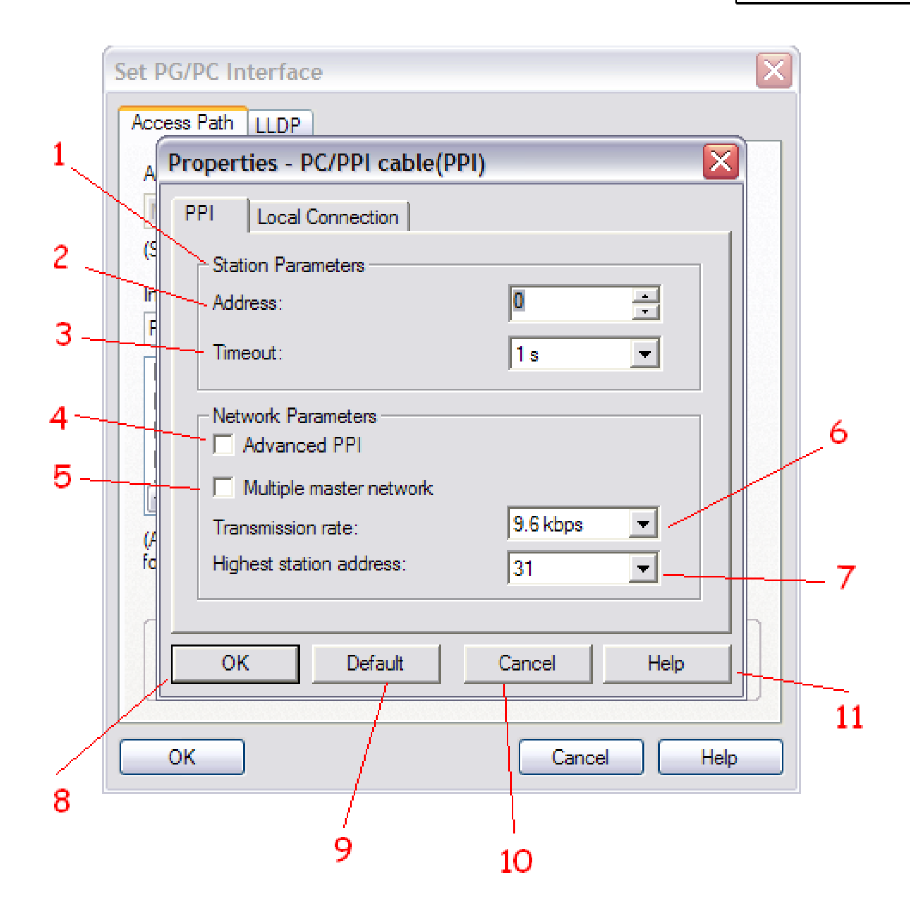
٣ - المقصود بــ Select لاختيار الطرق المراد استخدامها فى التوصيل (كابل – نيت – شريحة محمول)

٤- المقصود بــ Help هى صفحة تحتوى على شرح تفصيلى لصفحة الــ SET PG/PC.

٥- المقصود بــ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

٦- المقصود بــ Ok الخروج من الصفحة مع حفظ جميع التغيير.

صفحة الـ PPI :



١- المقصود بـ Station Parameters هي قأمة للمتغيرات الخاصة بوحدة الـ PLC.

٢- المقصود بـ Address هو العنوان الخاص بوحدة الـ PLC.

٣- المقصود بـ Timeout هو الوقت المحدد لظهور رسالة بعد إنقطاع الكابل.

٤- المقصود بـ Advanced PPI هو التحكم بالـ PLC باستخدام أكثر من مصدر.

٥- المقصود بـ Multiple master network في حالة توصيل أكثر من جهاز PLC معاً.

٦- المقصود بـ Transmission rate هي سرعة نقل المعلامات من و إلى الـ PLC.

٧- المقصود بـ Highest station address أقصى عدد من أجهزة الـ PLC المراد التحكم بها بواسطة الكمبيوتر.

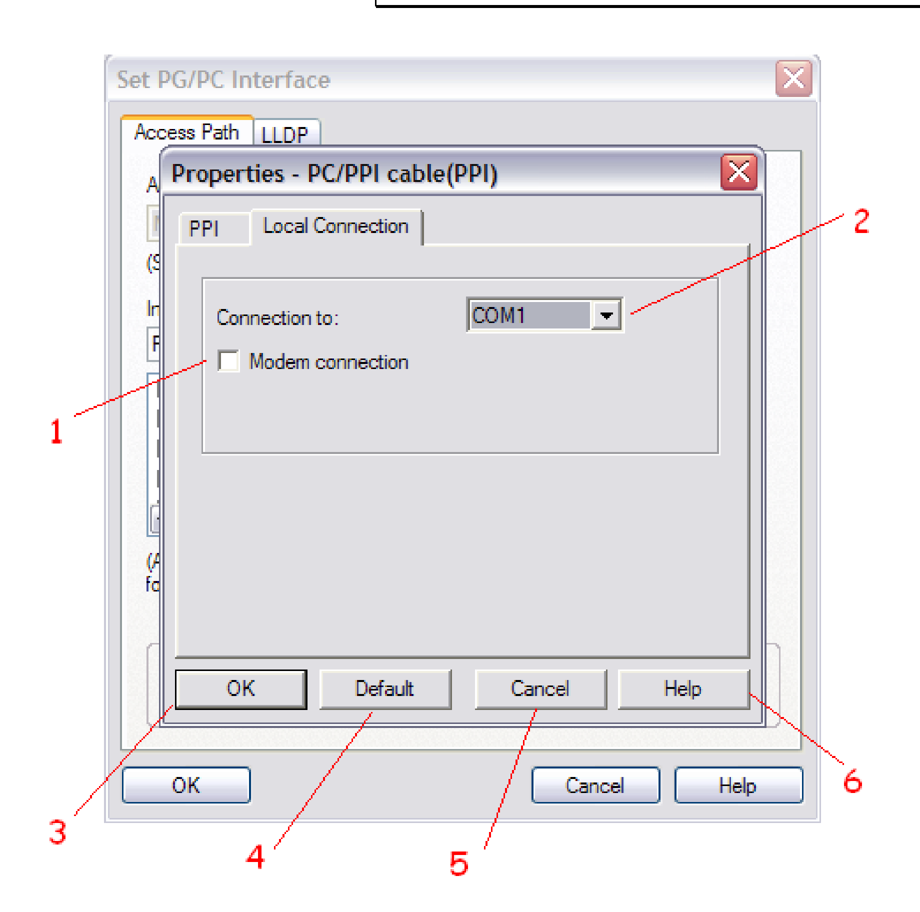
٨- المقصود بـ Ok الخروج من الصفحة مع حفظ جميع التغيير.

٩- المقصود بـ Default تستخدم للعودة المتغيرات إلى قيمتها الأصلية.

١٠- المقصود بـ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

١١- المقصود بـ Help هى صفحة تحتوى على شرح شامل و تفصيلى لصفحة الـ SET PG/PC.

صفحة الـ LOCAL CONNECTION:



١- المقصود بـ Connection to هو نوع طرف الكابل الموصل بالكمبيوتر. قد يكون: (USB أو COM)

٢- المقصود بـ Modem connection هو التحكم بوحدة الـ PLC بواسطة شبكة النت.

٣- المقصود بـ Ok الخروج من الصفحة مع حفظ جميع التغيير.

٤- المقصود بـ Default تستخدم للعودة المتغيرات إلى قيمتها الأصلية.

٥- المقصود بـ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

٦- المقصود بـ Help هى صفحة تحتوى على شرح شامل و تفصيلى لصفحة الـ SET PG/PC INTERFACE.

خطوات تحميل البرنامج:

١- يتم رسم البرنامج المراد تنفيذه كما سيوضح بالتفصيل فيما بعد.

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.1
I0.0
Q0.0
Q0.0
Network 2

٢- يتم الضغط على Compile all أو Compile لمعرفة أذا كانت توجد أخطاء فى الرموز المستخدمة.




```
Compiling Program Block...
MAIN (OB1)
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 24 (bytes), 0 errors

Compiling Data Block...
Block Size = 0 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

Total Errors: 0

Ready
```

ملاحظة: بالضغط على compile all أو compile:

- يظهر عدد الأخطاء كما فى الصورة السابقة بحيث يكتب Total Errors.
- فى حالة وجود أخطاء يكتب أيضاً مكان الرمز الخطاء و لماذا هو خطاء.
- بالضغط على الرسالة التى توضح الخطاء فأنه يظهر الرمز الخطاء تلقائيا.

٣- يتم الضغط على Stop للتأكد من أن الـــ PLC لا يعمل.



٤- يتم الضغط على Download لتحميل البرنامج من الـــ computer إلى الـــ PLC.

Download

PPI Connection 2

Use the Options button to select blocks to download. 3

1

Remote Address: 2 4

5 CPU 226 REL 01.21

Click Download to begin.

8

6 9 7 10 11 12

Options

Download

Cancel

Options

Program Block

Data Block

System Block

Recipes

Data Log Configurations

13

To : PLC

To : PLC

To : PLC

14

Click for Help and Support 15

Close dialog on success 16

Prompt on RUN to STOP 17

١- عنوان صفحة التحميل.

٢- طريقة التوصيل المستخدمة بين الــ computer و الــ PLC.

٣- وصف طريقة الاستخدام.

٤- عنوان الــ CPU.

٥- نوع الــ CPU.

٦- تعنى تنفيذ خطوات عملية التحميل دون أخطاء.

٧- المكان الذى يعرض فيه رسالة التحذير أو الخطاء.

٨- صورة متحركة لتوضيح عملية التحميل.

٩- يستخدم لإظهار أو أخفاء بعض الخيارات.

١٠-تظهر فقط فى حالة رسم البرنامج على CPU معينة وتحميله على CPU أخرى, ولكنها لا تظهر فى الصورة السابقة.

١١- بالضغط يبدأ عملية التحميل ولكن يجب أن لا تكون أى أخطاء.

١٢- لإلغاء عملية التحميل.

١٣- أختيار الأجزاء المراد تحميلها.

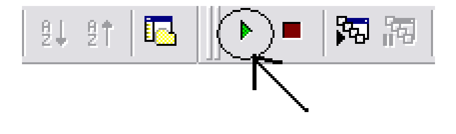
١٤- توضح أنه يتم نقل البرنامج إلى الــ PLC.

١٥- للمساعدة بحيث أنه يفتح صفحة help.

١٦- يغلق صفحة التحميل تلقائياً عند أنتهاء التحميل بنجاح.

١٧- نظراً لأنه لا يمكن تحميل البرنامج بينما الــ CPU فى وضع RUN فيقوم بتحويل الــ CPU من RUN إلى STOP.

٥- يتم الضغط على Run للتأكد من أن جهاز الــ PLC قد بدأ فى العمل وفى وقراءة البرنامج.



الباب الرابع

البرمجة

- لغـــــــــــــات البرمجة داخل جهاز الــ PLC.
- مــســميات الــمــدخــلات والــمــخــرجــات.
- لغـــــــــــــــة المخطط السلمى **(LAD)**.
- لغـــــــــــــــة مخطط البوبات المنطقية **(FBD)**.
- لغـــــــــــــــة قائمة الأجراءات **(STL)**.
- شـــــــــــــــرح لغة المخطط السلمى بالتفصيل.
- شـــــــــــــــرح الــ cycle time.
- شـــــــــــــــرح الــ scan time.
- شـــــرح كــيــفــيــة قــراءة الــبــرنــامـــج.
- تمـــــــــــــارين تطبيقية بأستخدام لغة ladder.

لغات البرمجة داخل جهاز الـــ PLC:

١- المخطط السلمى (Ladder Diagram Method):

٢- مخطط البوبات المنطقية (Function Block Diagram):

٣- قائمة الإجراءات (Statement List):

رقم	لغة البرمجة	اختصارها
١	Ladder Diagram Method	LAD
٢	Function Block Diagram	FBD
٣	Statement List	STL

- لغة الـــ LADDER هى أسهل اللغات فى البرمجة وذالك لأنها تشبه الكنترول كثيراً وهى التى سوف يتم التركيز عليها فى هذا الكتاب.

- لغة الـــ FBD هى ثانى أسهل اللغات فى البرمجة وذالك لأنها تشبه البوابات الإلكترونية كثيراً وسوف يتم شرح بعض التمارين بهذه اللغة فى الجزء الثانى من هذا الكتاب.

- لغة الـــ STL هى تعتبر أصعب اللغات فى البرمجة وذالك لأنها تتكون من كلمات ولكنها تتميز بحرية كتابة البرنامج دون تتبع ترتيب معين كما فى اللغتان الأخريتين وسوف يتم شرح بعض التمارين بهذه اللغة فى الجزء الثانى من هذا الكتاب.

مسميات المدخلات و المخرجات:

المدخلات:

مثال:

Ibit0.7 Ibit0.6 Ibit0.5 Ibit0.4 Ibit0.3 Ibit0.2 Ibit0.1 Ibit0.0

I Byte 0

مثال أخر:

Ibit1.7 Ibit1.6 Ibit1.5 Ibit1.4 Ibit1.3 Ibit1.2 Ibit1.1 Ibit1.0

I Byte 1

المخرجات:

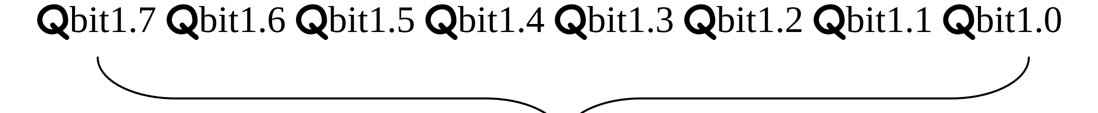
مثال:

Qbit0.7 Qbit0.6 Qbit0.5 Qbit0.4 Qbit0.3 Qbit0.2 Qbit0.1 Qbit0.0

Q Byte 0

مثال أخر:

Qbit1.7 Qbit1.6 Qbit1.5 Qbit1.4 Qbit1.3 Qbit1.2 Qbit1.1 Qbit1.0



Q Byte 1

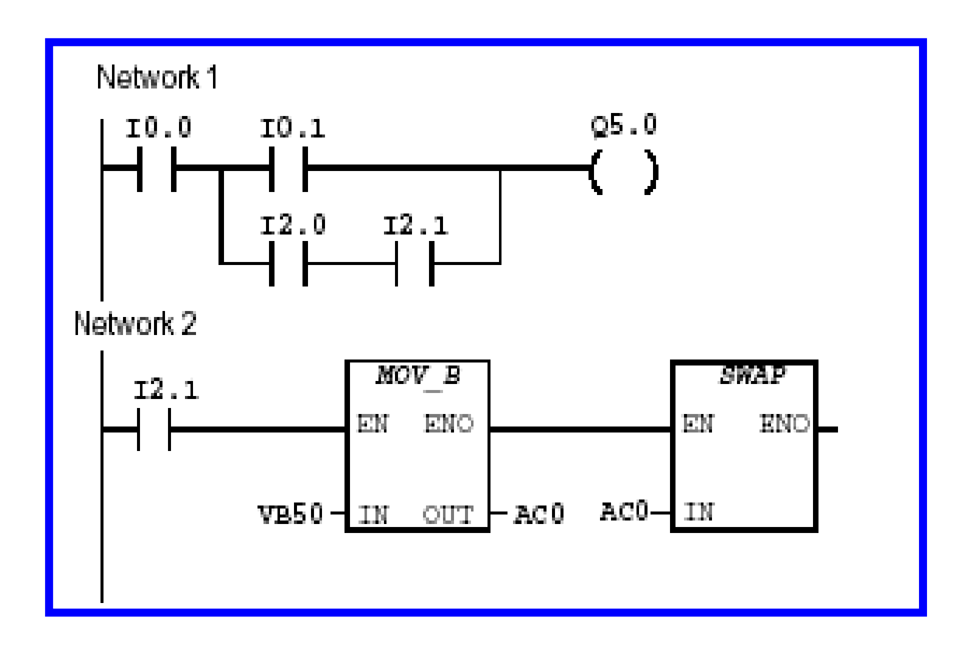
معلومات عن كل لغة:

المخطط السلمى (Ladder Diagram Method):

هذه الطريقة هى أقرب ما تكون لطريقة الكنترول المستخدم فى الدوائر الكهربية ودوائر التحكم و هى تعتبر من أسهل الطرق وأكثر الطرق استخداماً لعمل الدوائر الكهربية, مع ملاحظة أن الرسم يتم بطريقة أفقية.

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	رمز المخطط السلمى	الوظيفة
١	**S1**		**I0.3**		**مفتاح تشغيل**
٢	**S2**		**I1.7**		**مفتاح إيقاف**
٣	**H1**		**Q1.1**		**مخرجات**

- شكل عام للبرمجة بلغة الـ LADDER:

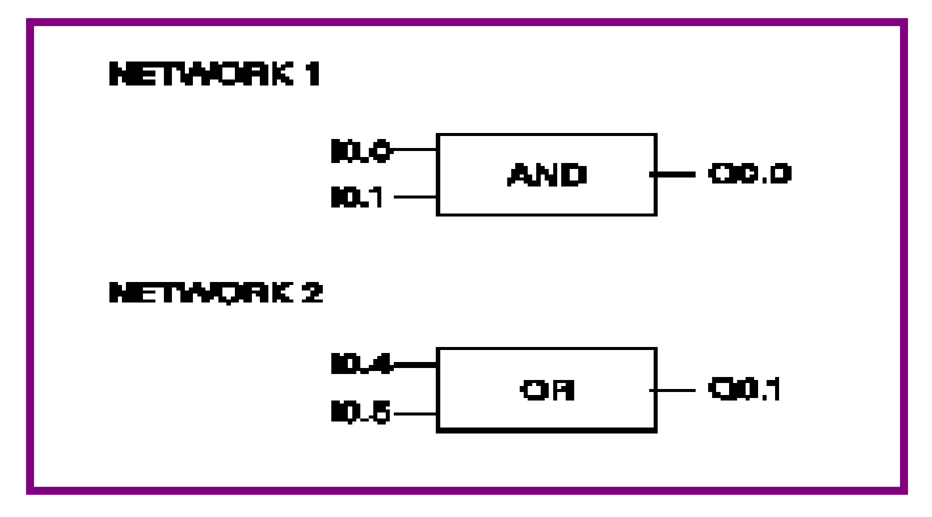


- **مخطط البوابات المنطقية (Function Block Diagram)**:

فى هذه الطريقة يتم استخدام البوابات المنطقية فى تنفيذ عمليات التحكم, وهى تعتبر من اللغات التى يسهل لمن يعمل فى مجال الإلكترونيات استخدامها, والبوابات المنطقية الأساسية الثلاث المستخدمة هى:

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	الوظيفة
١	أو	≥1	**OR**	مفتاحين على التوازى
٢	و	&	**AND**	مفتاحين على التوالى
٣	النفى		**NOT**	عكس الحالة

- شكل عام للبرمجة بلغة الـــ **FBD**:



- **قائمة الإجراءات (Statement List)**:

هذة الطريقة يتم فيها وصف الدائرة المراد التحكم بها, بمجموعة أوامر, وهى مجموعة من الأوامر يعبر عنها بحروف.

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	الوظيفة
١	و	**A**	**AND**	مفتاح على التوالى
٢	أو	**O**	**OR**	مفتاح على التوازى
٣	لا	**N**	**NOT**	عكس الحالة

- **شكل عام للبرمجة بلغة الـــ STL**:

```
NETWORK 1
            LD          I0.0
            A           I0.1
            =           Q0.0
NETWORK 2
            LD          I0.4
            O           I0.5
            =           Q0.1
```

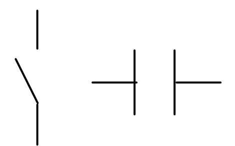
شرح لغة الـــ Ladder Diagram Method:

نلاحظ أن ─┤ ├─ هو مفتاح يبدو مفتوحاً و يستخدم للتشغيل و فى نفس الوقت يبدو أن هذا المفتاح ─┤/├─ هو مفتاح مغلقاً و يستخدم للإيقاف ولكن هذا غير صحيح لأن لمعرفة أذا كان المفتاح مغلق أو مفتوح يعتمد هذا على نوع المفتاح الموصل على جهاز الـــ PLC .

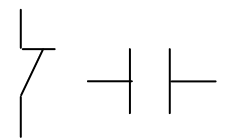
هذا المفتاح ─┤ ├─ له نفس حالة المفتاح الموصل على جهاز الـــ PLC من الخارج.

هذا المفتاح ─┤/├─ له عكس حالة المفتاح الموصل على جهاز الـــ PLC من الخارج.

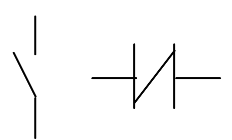
ملاحظة: هذا الشرح يخص فقط الــ inputs وليس أى contact.



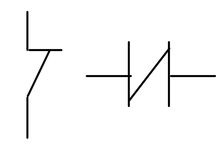
هو يعتبر مفتاح مفتوح (مفتاح تشغيل - normally open) لأنة له نفس حالة المفتاح الخارجى.



هو يعتبر مفتاح مغلق (مفتاح إيقاف - normally close) لأنة له نفس حالة المفتاح الخارجى.



هو يعتبر مفتاح مغلق (مفتاح إيقاف - normally close) لأنة له حالة عكس حالة المفتاح الخارجى.



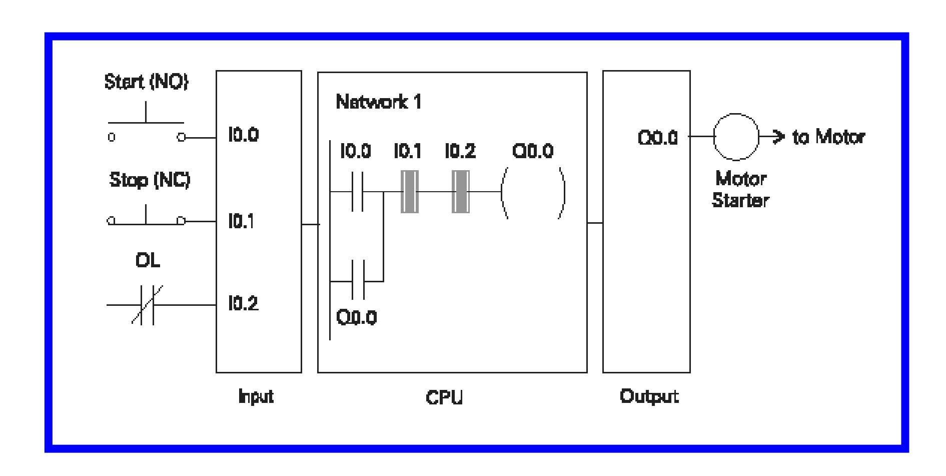
هو يعتبر مفتاح مفتوح (مفتاح تشغيل - normally open) لأنة له حالة عكس حالة المفتاح الخارجى.

مثال تجريبى: لمحرك يعمل من مكان واحد و يتوقف من مكان واحد مع وجود حماية overload.

شرح التمرين فى خمس خطوات:

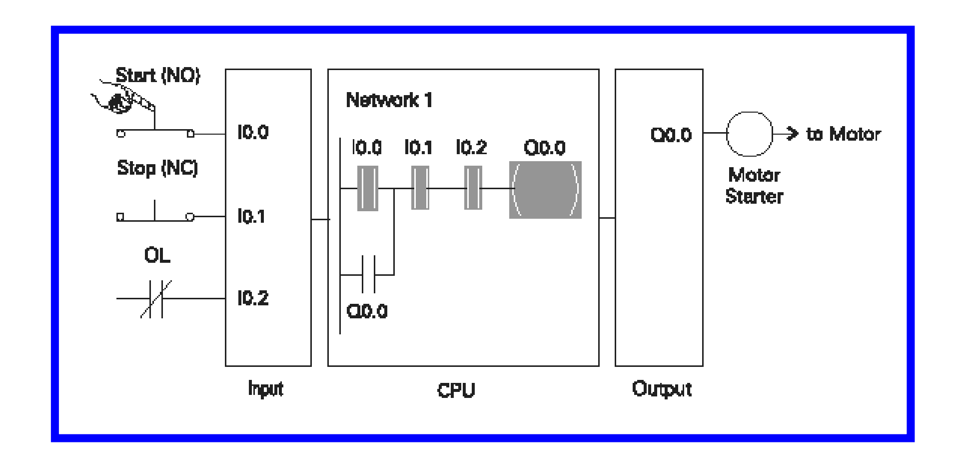
التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.



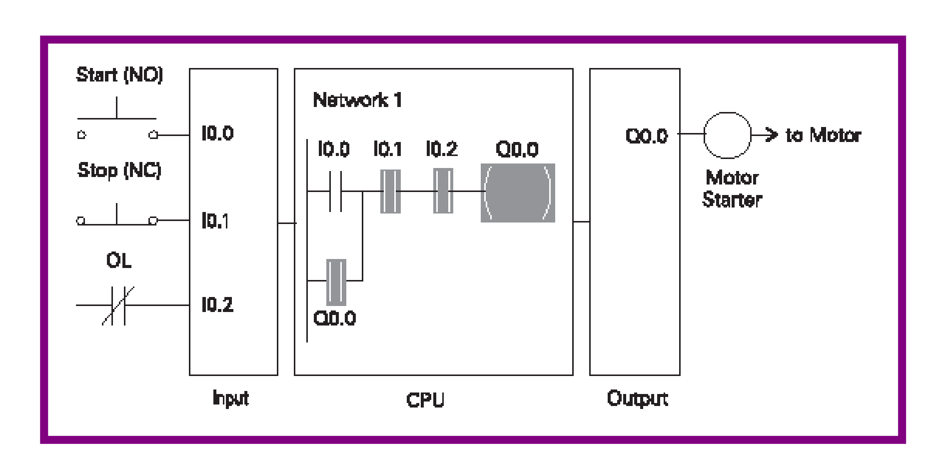
التمرين فى حالة تشغيل.

سوف نلاحظ أن I0.0 & I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.



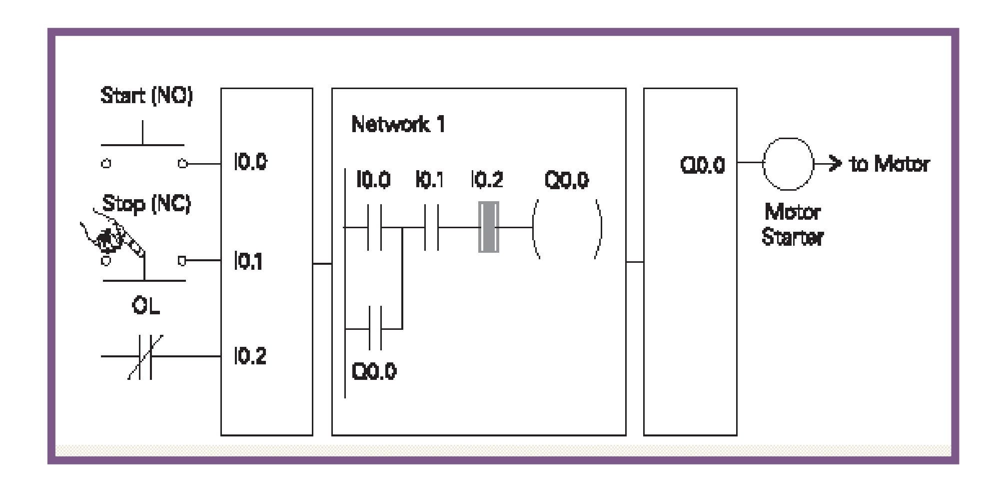
التمرين فى حالة تشغيل.

سوف نلاحظ أن نقطة التعويض أصبحت مغلقة.



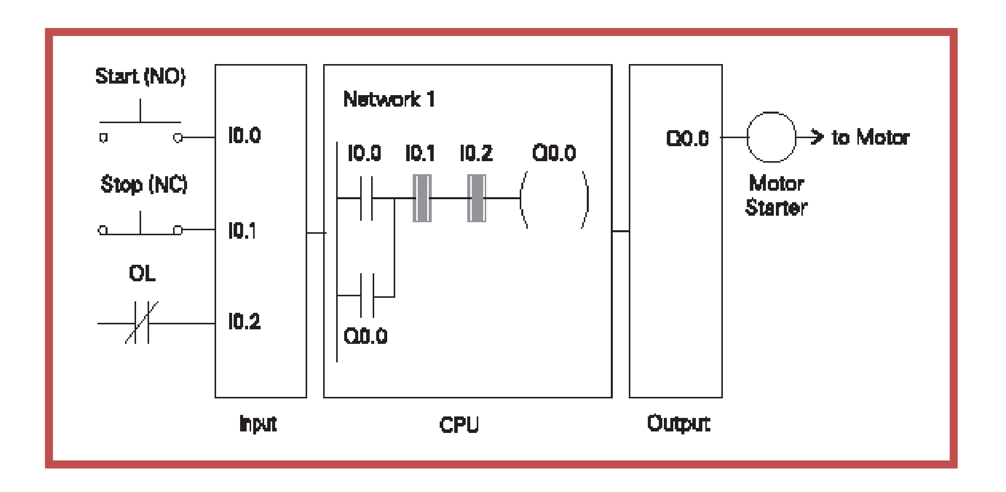
التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 له نفس الحالة للمفتاح الذى بالخارج أى مفتوح.



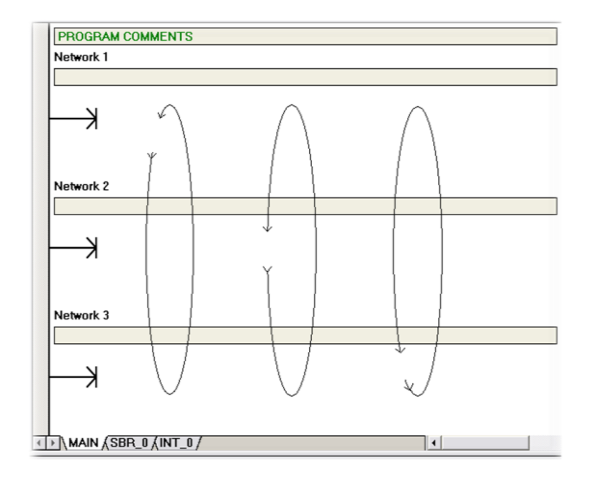
التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.



لمعرفة البرمجة فى الـــ **PLC** يجب معرفة:

- **أولاً:** Cycle time.
- **ثانياً:** Scan time.



الـــ Cycle time:

هو الـــزمن المستغـــرق لكـــي يقوم الـــ PLC بحلقة مغلقة كاملة أى بالتحرك من أى Network إلى أن يعـــود إلى نفس الـــ Network مرة أخـــرى ويتراوح الزمـــــــــن من 0.3ms إلى 1ms.

الــ Scan time :

هو الوقت اللازم لوحدة الــ PLC لكى:

١- يقراء المداخل (Read input).

٢- ينفذ البرنامج (Execute program).

٣- تعديل المخرجات (Update output).

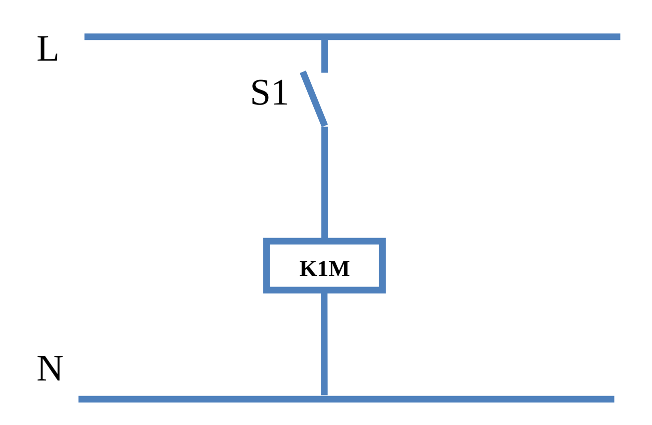
القانون: Scan time = Cycle time x Program size [Kbps]

ملاحظة: تتم قرأه البرنامج فى اتجاه واحد فقط من أعلى إلى أسفل و ليس العكس.

للتوضيح سوف نقوم ببعض الأمثلة:

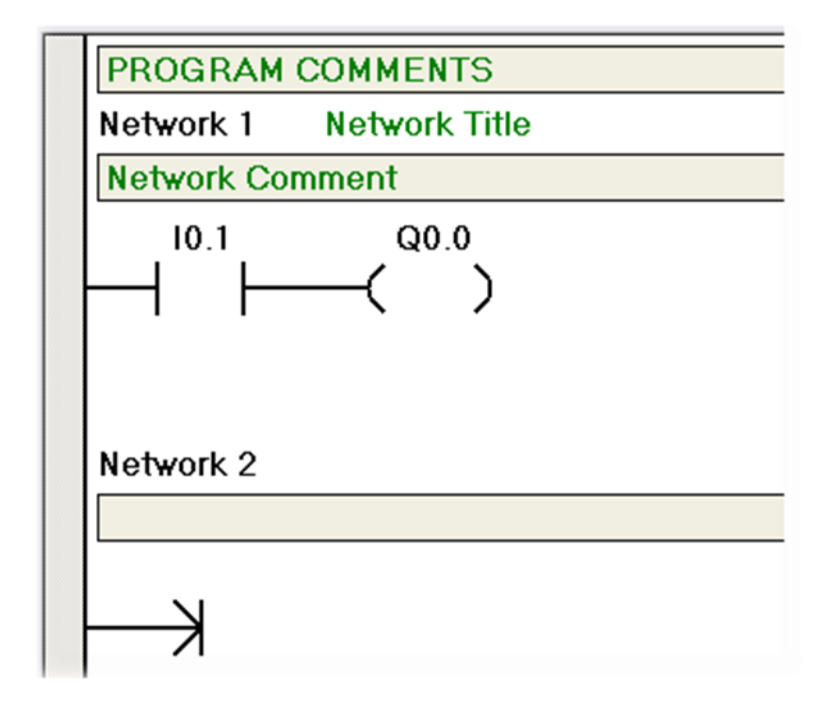
المثال الأول:

✓ محرك يعمل بواسطة مفتاح واحد:



عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.o.	I0.1/S1
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

البرنامج:

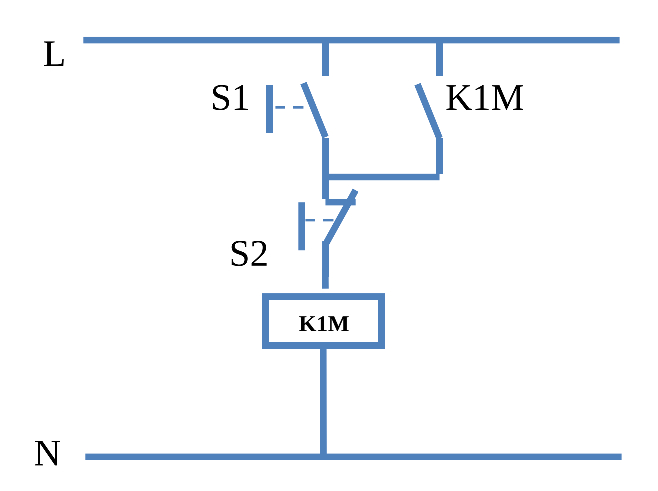


الشرح:

علماً بأن حالة المفتاح I0.1 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110) وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

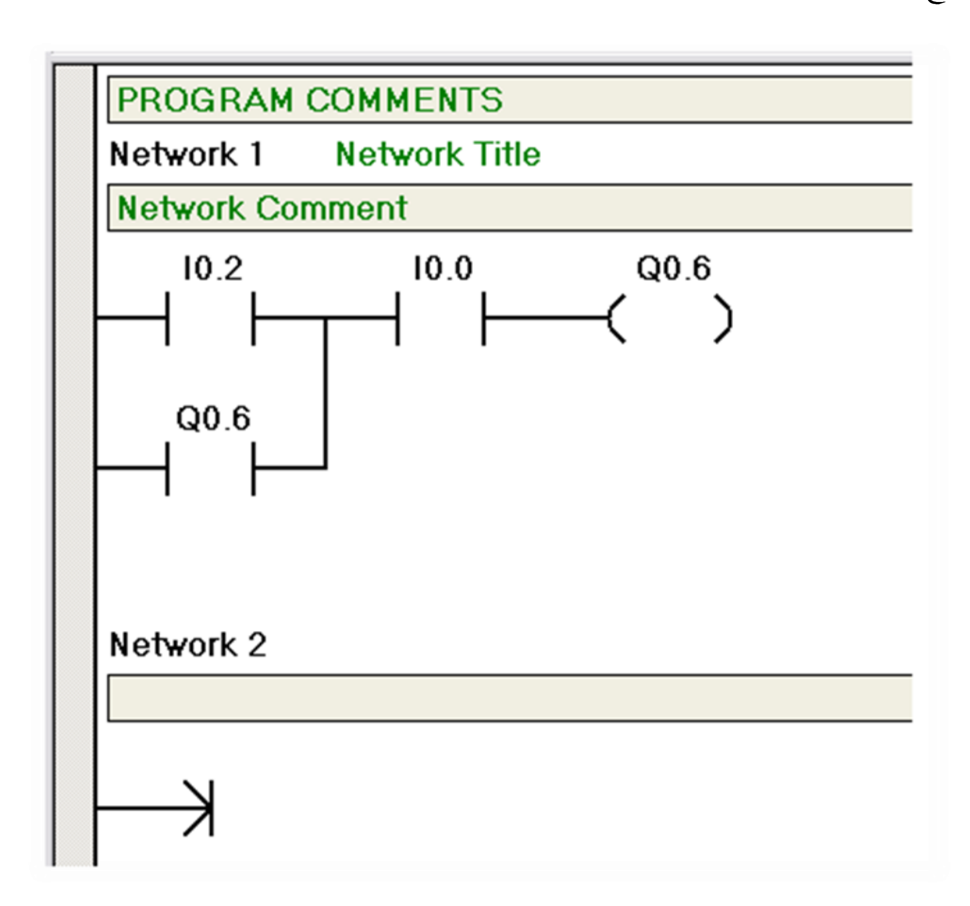
المثال الثانى:

✓ محرك يعمل بواسطة مفتاح واحد ويقف بواسطة مفتاح واحد:



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.2/S1
٢	n.c.	I0.0/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.6/K1M

البرنامج:



الشرح:

علماً بأن حالة المفتاح I0.1 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

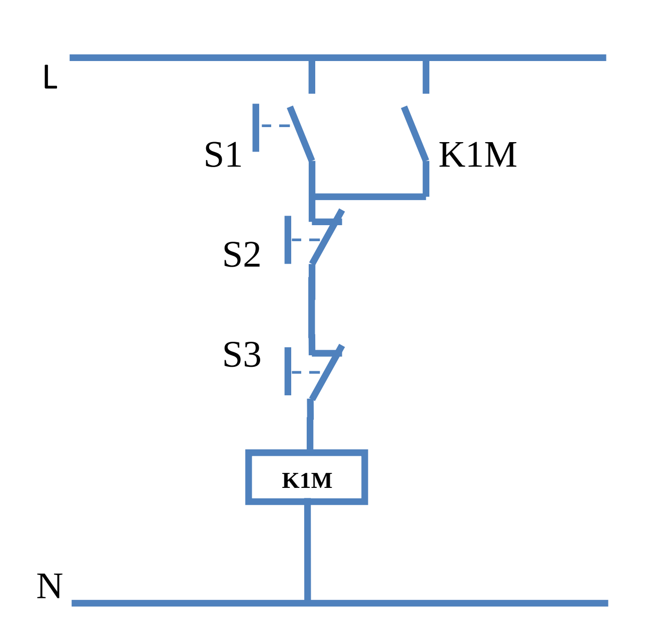
و حالة المفتاح I0.0 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

وبالضغط على المفتاح I0.0 بالخارج يفتح المفتاح أيضاً فى الداخل و يقف الخرج Q0.0

المثال الثالث:

✓ محرك يعمل بواسطة مفتاح واحد ويقف بواسطة مفتاحين:



أسم الدخل	نوع الدخل	عدد الدخل
I0.5/S1	n.o.	١
I0.6/S2	n.c.	٢
I0.7/S3	n.c.	٣
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.4/K1M	كونتاكتور	١

البرنامج:

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.5 I0.6 I0.7 Q0.4
Q0.4
Network 2

الشرح:

علماً بأن حالة المفتاح I0.5 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

و حالة المفتاح I0.6 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

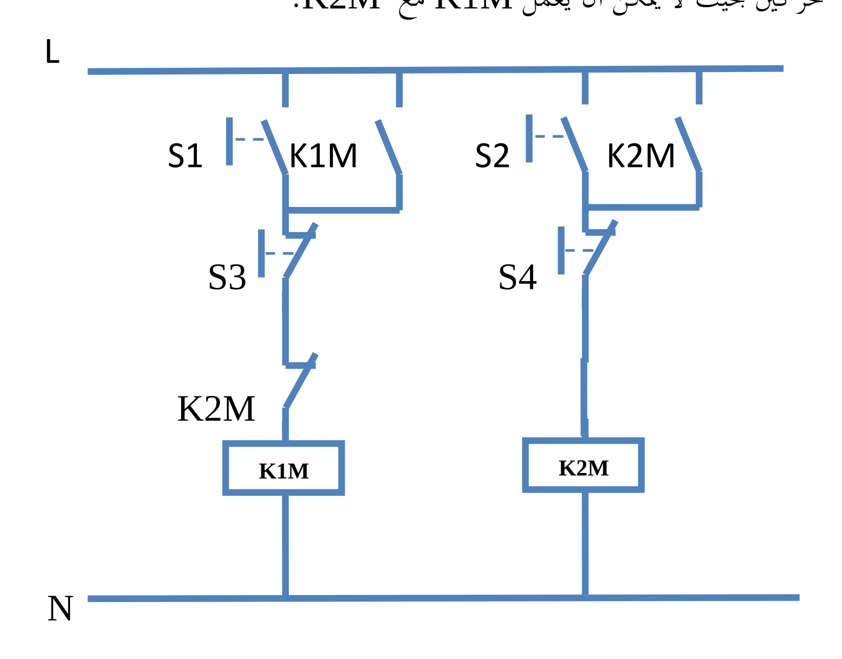
و حالة المفتاح I0.7 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.5 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

وبالضغط على المفتاح I0.6 أو I0.7 بالخارج يفتح المفتاح أيضاً فى الداخل و يقف الخرج Q0.0

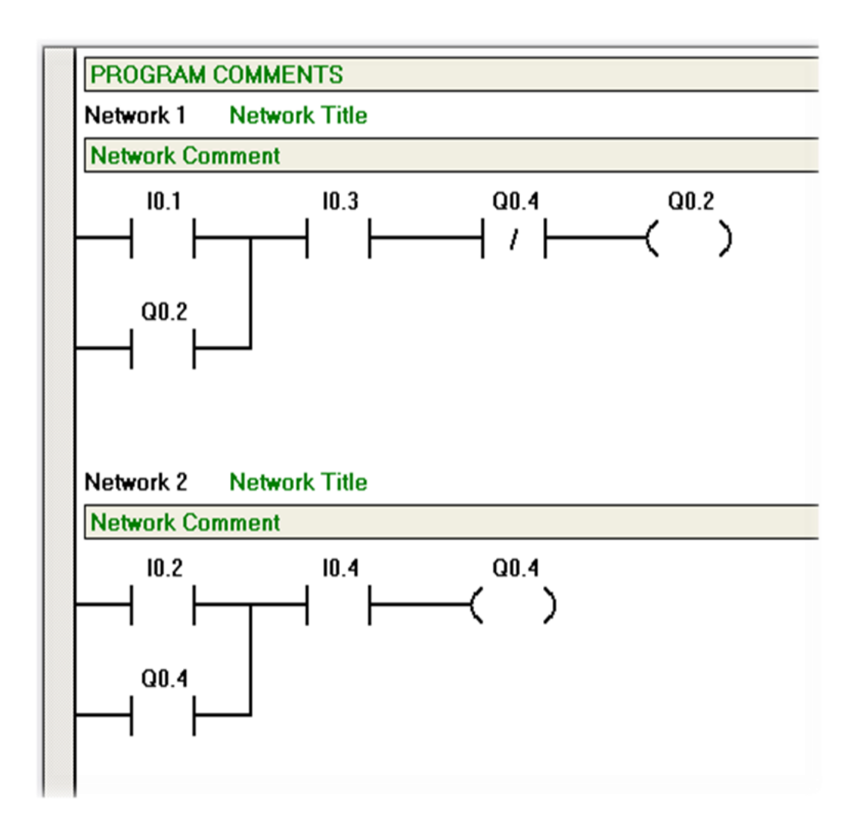
المثال الرابع:

✓ محركين بحيث لا يمكن أن يعمل K1M مع K2M:



أسم الدخل	نوع الدخل	عدد الدخل
I0.1 / S1	n.o.	١
I0.2 / S2	n.o.	٢
I0.3 / S3	n.c.	٣
I0.4 / S4	n.c.	٤
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١
Q0.4 / K2M	كونتاكتور	٢

البرنامج:



الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

وحالة المفتاح I0.3 و حالة المفتاح I0.4 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

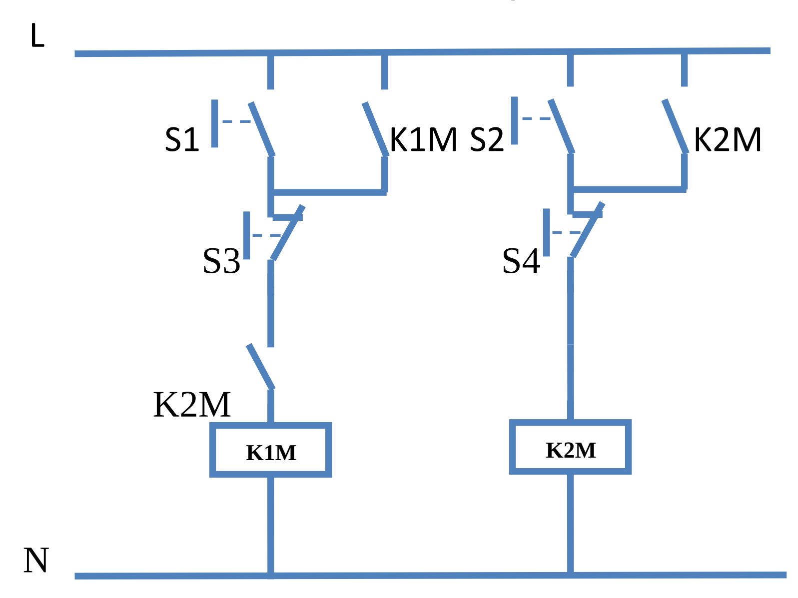
وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.2

وبالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.4

ولكن لا يمكن للإثنين أن يعملا معاً حيث أنه لا يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4

المثال الخامس:

✓ محركين بحيث لا يمكن أن يعمل K1M دون K2M



أسم الدخل	نوع الدخل	عدد الدخل
I0.1 / S1	n.o.	١
I0.2 / S2	n.o.	٢
I0.3 / S3	n.c.	٣
I0.4 / S4	n.c.	٤
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١
Q0.4 / K2M	كونتاكتور	٢

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I0.3 Q0.4 Q0.2

Q0.2

Network 2 Network Title

Network Comment

I0.2 I0.4 Q0.4

Q0.4

الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

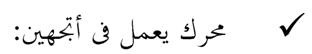
وحالة المفتاح I0.3 و حالة المفتاح I0.4 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل ولكن لا يمكن للخرج Q0.2 أن يعمل بسبب وجود نقطة مفتوحة من Q0.4

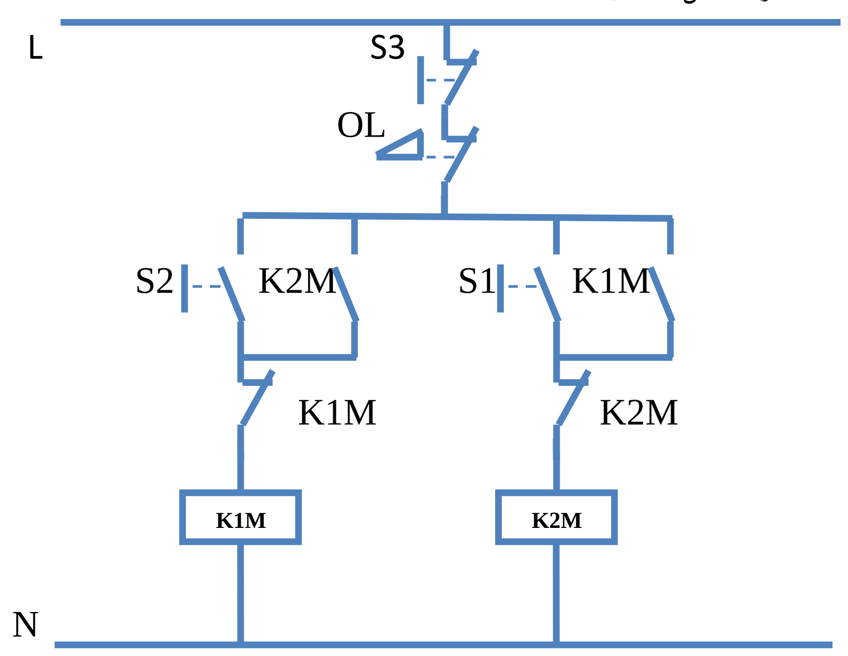
بينما بالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.4 دون الاعتماد على أى شروط أخرى

ولهذا يمكن للاثنين أن يعملا معاً حيث أنه يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4 وليس بدونه.

المثال السادس:



✓ محرك يعمل فى أتجهين:



أسم الدخل	نوع الدخل	عدد الدخل
I0.1 / S1	n.o.	١
I0.2 / S2	n.o.	٢
I1.1 / S3	n.c.	٣
I1.2 / OL	n.c.	٤
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١
Q0.4 / K2M	كونتاكتور	٢

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I1.1 I1.2 Q0.4 / Q0.2 ( )

Q0.2

Network 2 Network Title

Network Comment

I0.2 I1.1 I1.2 Q0.2 / Q0.4 ( )

Q0.4

الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

وحالة المفتاح I1.1 و حالة المفتاح I1.2 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل ويعمل الخرج Q0.2

وبالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل ويعمل الخرج Q0.4

ولكن لا يمكن للاثنين أن يعملا معاً حيث أنه توجد نقطة مفتوح من كل خرج فى طريق الخرج الأخر

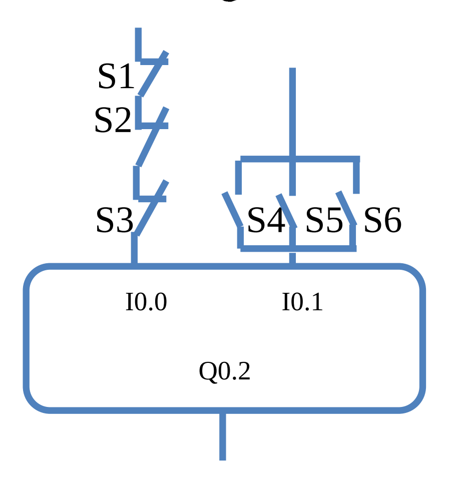
ولهذا لا يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4.

مثال أخر:

✓ محرك يعمل من ثلاث أماكن ويطفئ من ثلاث أماكن:

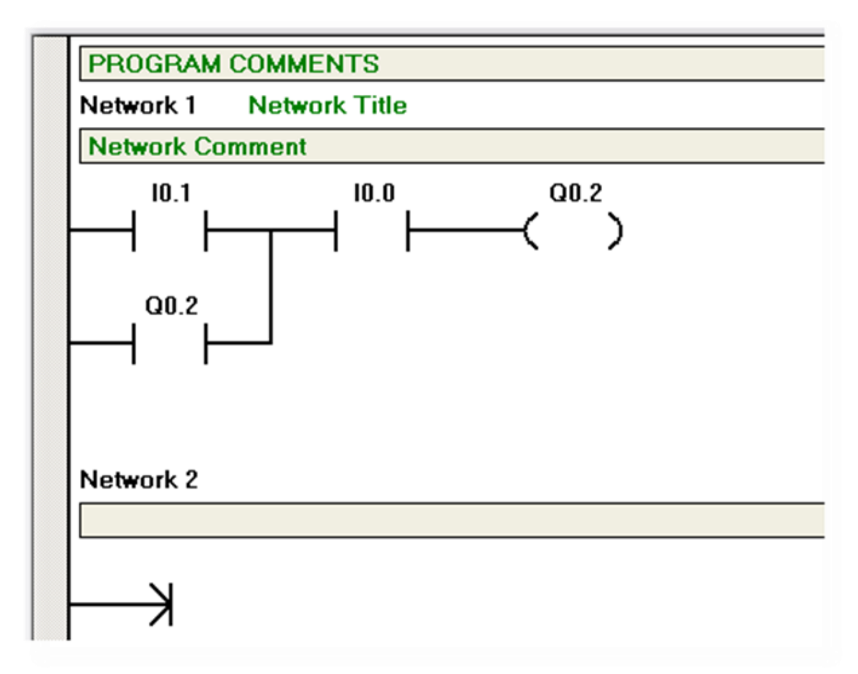
ملاحظة: بدلان من توصيل ستة مفاتيح على جهاز الـــ PLC يمكن توصيل كل ثلاث مفاتيح على نقطة واحدة, سواء كان بالتوالى أو بالتوازى:

فى هذه الحالة سيتم استخدام المفاتيح المغلقة للفصل والمفاتيح المفتوحة للتشغيل.



أسم الدخل	نوع الدخل	عدد الدخل
I0.0 / S1	n.c.	١
I0.0 / S2	n.c.	٢
I0.0 / S3	n.c.	٣
I0.1 / S4	n.o.	٤
I0.1 / S5	n.o.	٥
I0.1 / S6	n.o.	٦
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١

البرنامج:



الشرح:

علماً بأن حالة المفاتيح الموجودة على الدخل I0.1 فى الخارج هى مفاتيح مفتوحة وهى كذلك أيضاً فى البرنامج (أنظر صفحة 110).

وحالة المفاتيح الموجودة على الدخل I0.0 فى الخارج هى مفاتيح مغلقة وهى كذلك أيضاً فى البرنامج.

فبالضغط على أى من المفاتيح الموصلة بالدخل I0.1 يغلق المفتاح فى البرنامج و يعمل الخرج Q0.2

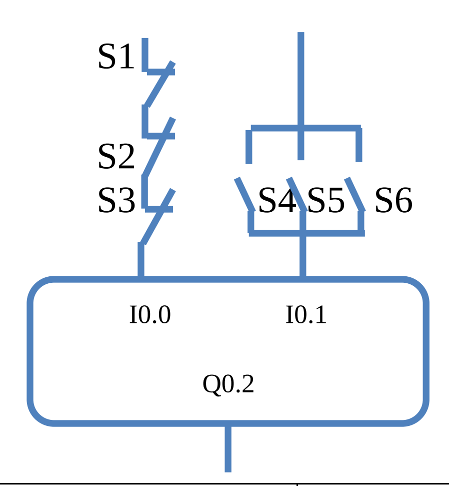
وبالضغط على أى من المفاتيح الموصلة بالدخل I0.0 يفتح المفتاح فى البرنامج و يقف الخرج Q0.2

مثال أخر:

- ✓ محرك يعمل من ثلاث أماكن ويطفأ من ثلاث أماكن:

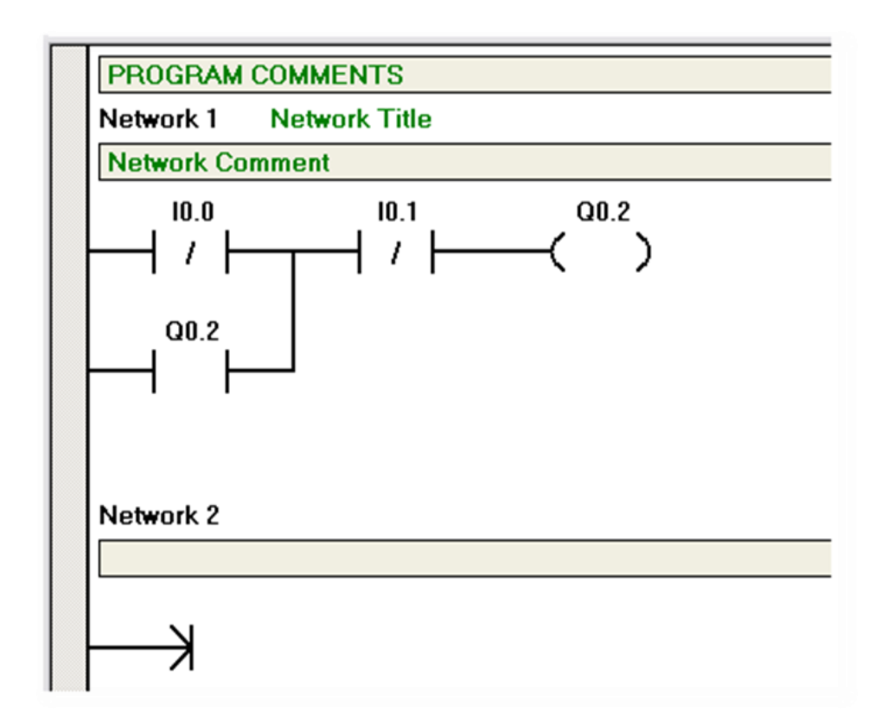
ملاحظة: بدلان من توصيل ستة مفاتيح على جهاز الـــ PLC يمكن توصيل كل ثلاث مفاتيح على نقطة واحدة, سواء كان بالتوالى أو بالتوازى:

في هذه الحالة سيتم استخدام المفاتيح المغلقة للتشغيل والمفاتيح المفتوحة للفصل.



أسم الدخل	نوع الدخل	عدد الدخل
I0.0 / S1	n.o.	١
I0.0 / S2	n.o.	٢
I0.0 / S3	n.o.	٣
I0.1 / S4	n.c.	٤
I0.1 / S5	n.c.	٥
I0.1 / S6	n.c.	٦
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١

البرنامج:



الشرح:

علماً بأن حالة المفاتيح الموجودة على الدخل I0.1 فى الخارج هى مفاتيح مفتوحة ولكنها مغلقة فى البرنامج بسبب استخدام مفتاح عكس الحالة (أنظر صفحة 110).

وحالة المفاتيح الموجودة على الدخل I0.0 فى الخارج هى مفاتيح مغلقة ولكنها مفتوحة فى البرنامج بسبب استخدام مفتاح عكس الحالة.

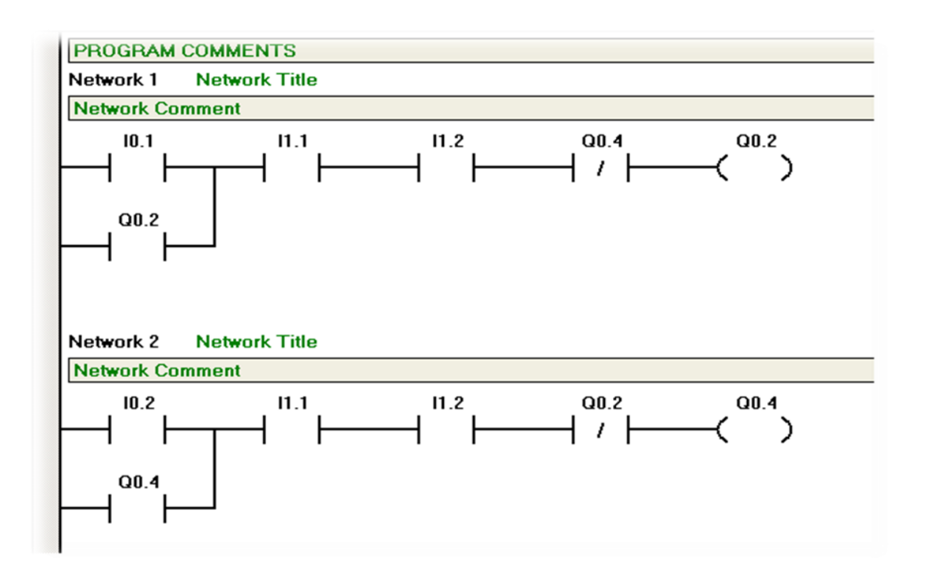
فبالضغط على أى من المفاتيح الموصلة بالدخل I0.0 تقطع الإشارة فى الخارج ولكن يغلق المفتاح فى البرنامج بسبب عكس الحالة و يعمل الخرج Q0.2

وبالضغط على أى من المفاتيح الموصلة بالدخل I0.1 ترسل إشارة إلى وحدة الــ PLC من الخارج ولكن يفتح المفتاح فى البرنامج بسبب عكس الحالة و يقف الخرج Q0.2

لمعرفة طريقة قراءة الـــ PLC لأى برنامج فلنستخدم كمثال للشرح, التمرين الذى رسمناه فى الصفحات السابقة وهو محرك اتجاهين.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1 / S1
٢	n.o.	I0.2 / S2
٣	n.c.	I1.1 / S3
٤	n.c.	I1.2 / S4

البرنامج:



فى حالة الضغط على المفتاح I0.2:

رقم الــ CYCLE	Network	ماذا يحدث
Cycle رقم (n)	Network1	لا يحدث أى شيء لأن Q0.4 لم تعمل بعد.
	Network2	تصبح جميع النقط مغلقة وتعمل Q0.4 وتغلق أيضاً نقطة التعويض.
Cycle رقم (n+1)	Network1	سوف تفتح النقطة المغلقة لــ Q0.4.
	Network2	تبقى تعمل كما هى

فى حالة فتح المفتاح I0.2:

رقم الــ CYCLE	Network	ماذا يحدث
Cycle رقم (n+3000)	Network1	سوف تبقى نقطة Q0.4 مفتوحة.
	Network2	تبقى Q0.4 تعمل لأن نقطة التعويض مغلقة.
Cycle رقم (n+3001)	Network1	سوف تبقى نقطة Q0.4 مفتوحة.
	Network2	تبقى تعمل كما هى

كيف سوف يعمل هذا التمرين إذا قمنا:

١. بفتح مفتاح الإيقاف الخاص بالاتجاهين I1.1

٢. بغلق مفاتيح التشغيل I0.1 و I0.2

٣. ثم نقوم بعد ذلك بغلق مفتاح الإيقاف الخاص بالاتجاهين I1.1

الإجابة:

نظراً إلى أن وحدة الــ PLC تقرأ البرنامج بنظام وترتيب فأن الوحدة ستمر أولاً على Q0.2 ثم تمر على Q0.4 ولهذا سوف تعمل Q0.2 ولهذا لا يمكن للخرج Q0.4أن يعمل فى هذه الحالة بسبب وجود نقطة مغلقة من الخرج الأول Q0.2 فلى طريق الخرج الثانى Q0.4.

الريليه...... Markers :

الــ marker هو ريليه يستخدم داخل وحدة الــ PLC فى البرمجة.

المسميات الخاصة بالــ markers هى:

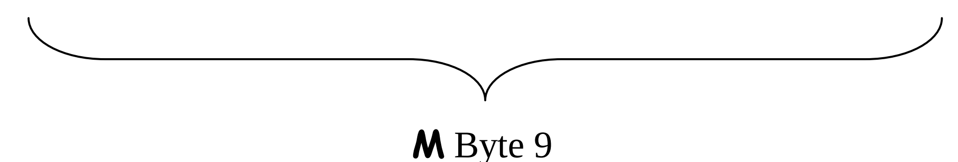
مثال:

Mbit0.7 Mbit0.6 Mbit0.5 Mbit0.4 Mbit0.3 Mbit0.2 Mbit0.1 Mbit0.0



مثال أخر:

Mbit9.7 Mbit9.6 Mbit9.5 Mbit9.4 Mbit9.3 Mbit9.2 Mbit9.1 Mbit9.0



M Byte 9

الشكل:

M0.4

مثال:

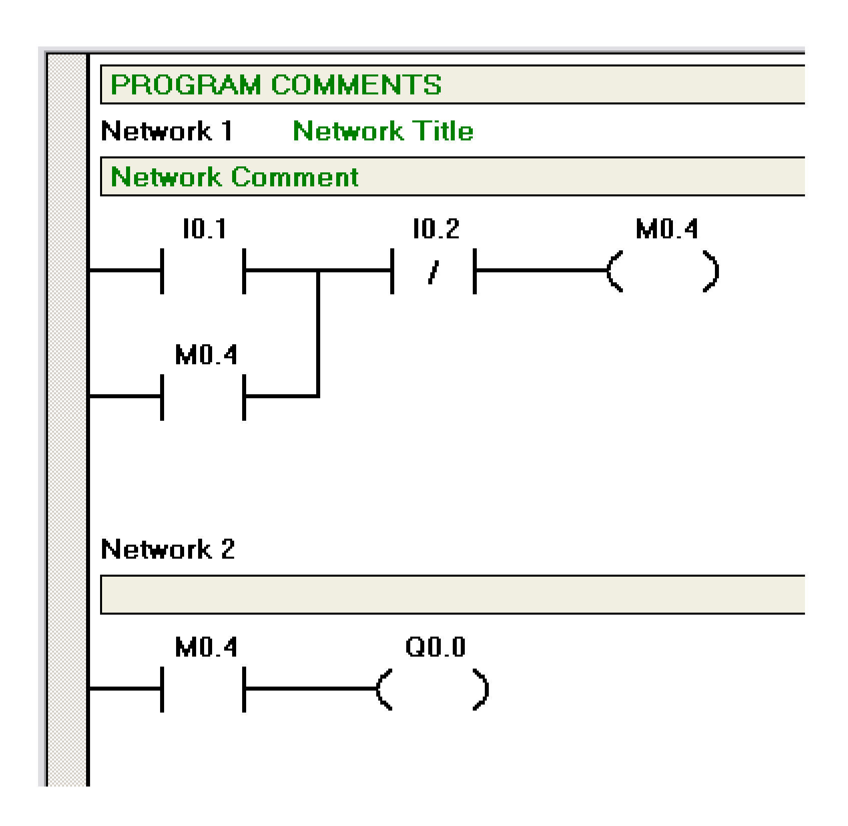
✓ محرك يعمل من مكان واحد و يقف من مكان واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

ملاحظة:

١- تم عكس حالة المفتاح I0.2 لأنة مفتوح بالخارج.
٢- الـــ marker يستخدم كمساعد داخل الـــ PLC ولا يمكن استخدامه كخرج.
٣- استخدام الـــ marker فى هذا التمرين ليس عملياً ولكن تم استخدامه فقط للتوضيح.
٤- يمكن استخدام الـــ marker مع الـــ Set/Reset أيضاً كما سيتم شرحها بعد قليل.

البرنامج:



الشرح:

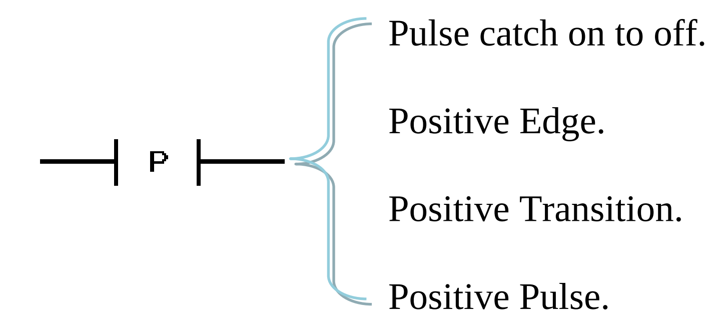
:Network1

بالضغط على I0.1 يعمل M0.4 وتغلق نقطة الحفظ بحيث في حالة فتح I0.1 يستمر الـــ marker في العمل.

:Network2

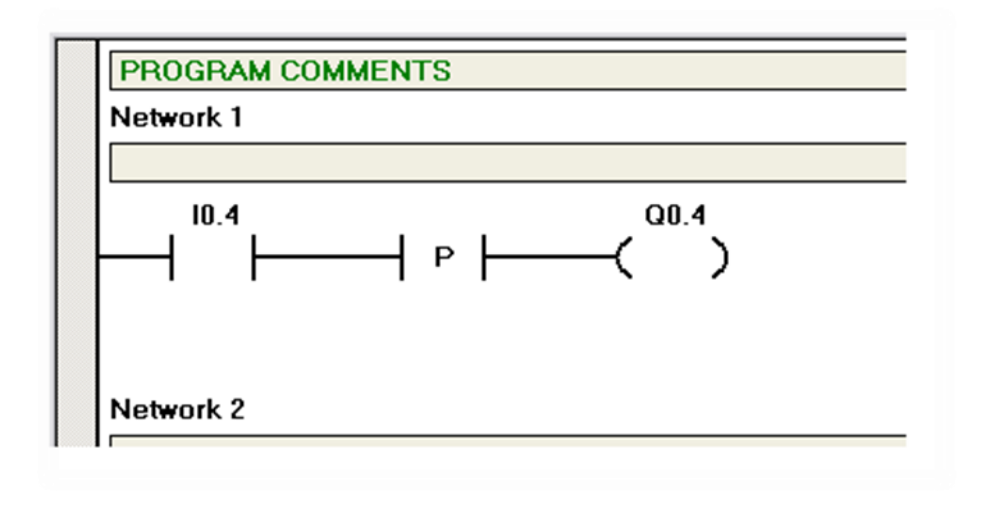
بعد أن يعمل M0.4 يعمل في نفس الـــ Cycle أيضاً الخرج Q0.0

مفتاح الـــ Positive edge :

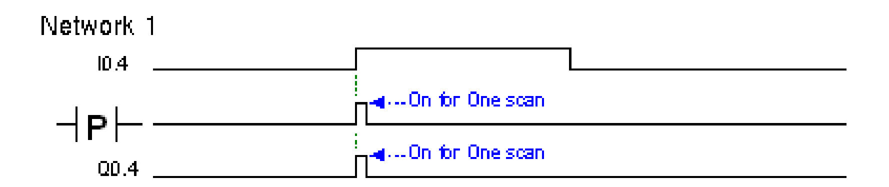


كل هذه المسميات السابقة هى لنفس الرمز.

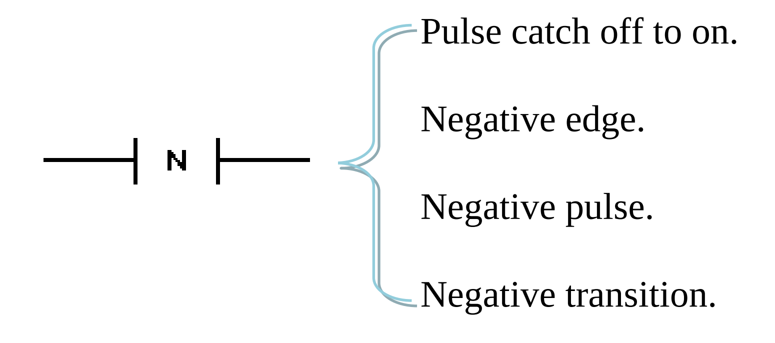
مفتاح الـــ positive edge هو مفتاح يوصل بالتوالى بعد أى مفتاح أخر بحيث عندما نقوم بغلق المفاتيح التى تسبق مفتاح الـــ positive edge فيوصل المفتاح الإشارة لزمن يعادل زمن الـــ cycle time ولتكرير هذه الإشارة يجب فتح أى مفتاح من الذين يسبقوا الـــ positive edge ثم غلقه مرة أخرى.



رسم تخطيطى:



مفتاح الـــ Negative edge :



كل هذه المسميات السابقة هى لنفس الرمز.

مفتاح الـــ Negative edge هو مفتاح يوصل بالتوالى بعد أى مفتاح أخر بحيث عندما نقوم بغلق المفاتيح التى تسبق مفتاح الـــ Negative edge فلا يوصل المفتاح الإشارة ولكن عند فتح أى أو كل المفاتيح التى تسبق مفتاح الـــ Negative edge فيوصل المفتاح الإشارة لزمن يعادل زمن الـــ cycle time ولتكرير هذه الإشارة يجب غلق أى مفتاح من الذين يسبقوا الـــ Negative edge ثم فتحة مرة أخرى.

البرنامج

PROGRAM COMMENTS

Network 1

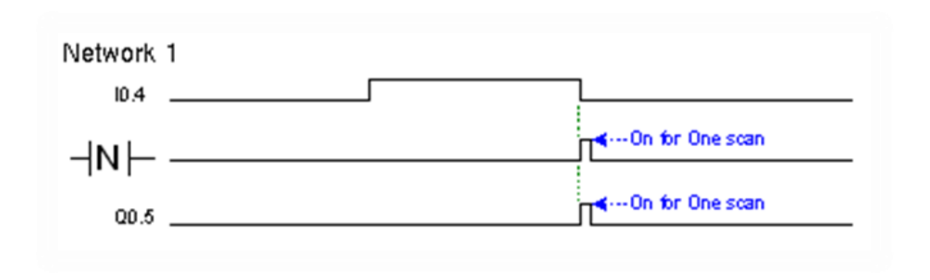
I0.4

N

Q0.5

Network 2

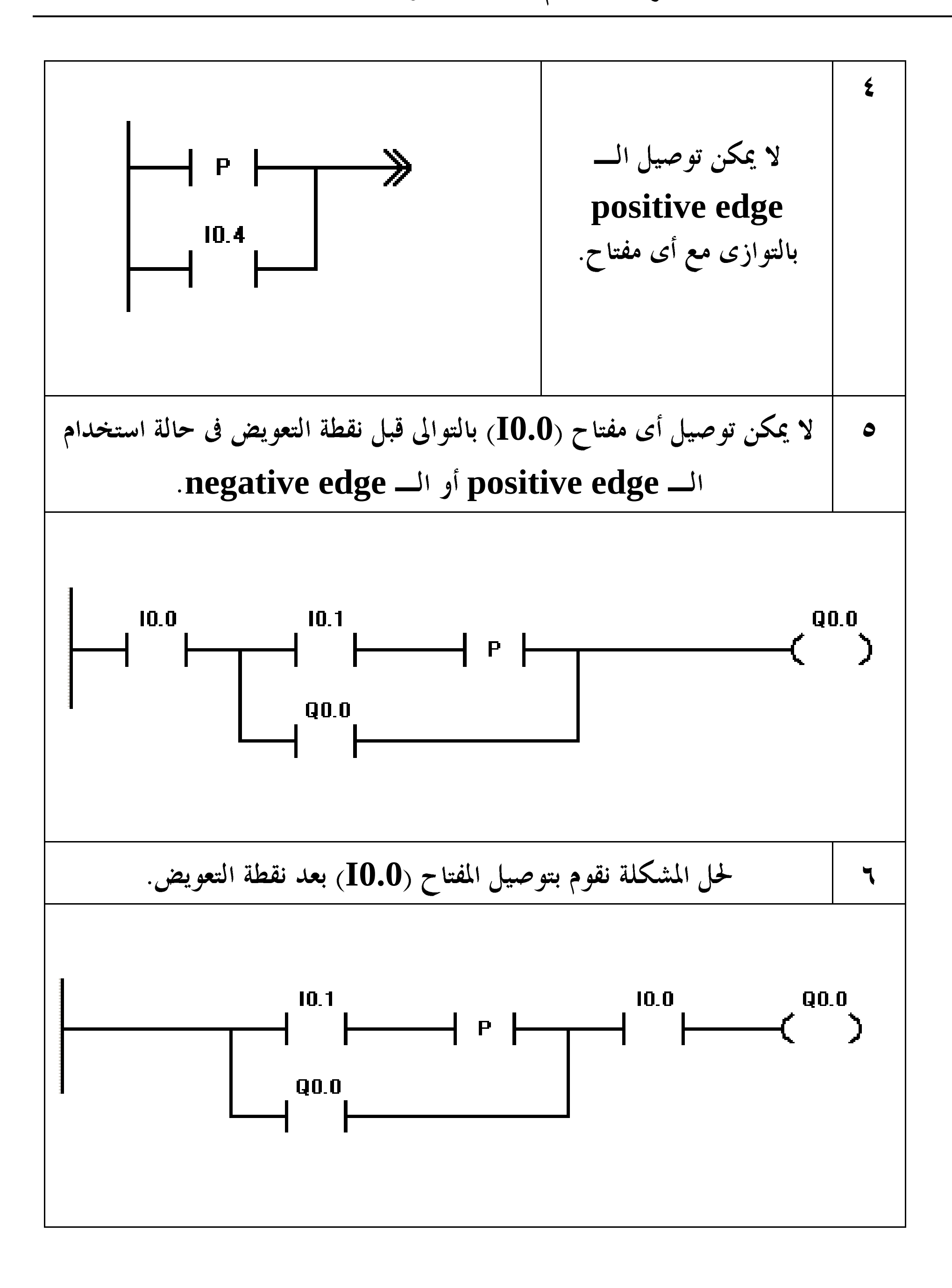
رسم تخطيطي:



ملاحظة:

توجد بعض التوصيلات الخاصة بالــ positive edge و الــ negative edge التى لا يجب تصميمها وسوف يتم توضيحها فى الجدول التالى لتجنب الأخطاء.

م	الشرح	الرسم
١	**لا يمكن توصيل الــ positive edge دون توصيل مفتاح أولاً.**	P
٢	**لا يمكن توصيل الــ negative edge دون توصيل مفتاح أولاً.**	N
٣	**لا يمكن توصيل الــ negative edge بالتوازى مع أى مفتاح.**	N Q0.6 /



٤	لا يمكن توصيل الــ **positive edge** بالتوازى مع أى مفتاح.	
٥	لا يمكن توصيل أى مفتاح (**I0.0**) بالتوالى قبل نقطة التعويض فى حالة استخدام الــ **positive edge** أو الــ **negative edge**.	
٦	لحل المشكلة نقوم بتوصيل المفتاح (**I0.0**) بعد نقطة التعويض.	

الـــ Set/Reset .

رقم	الأسم	الرمز
١	Set	??.? —(S) ????
٢	Reset	??.? —(R) ????

الـــ Set:

??.?
—(S)
????

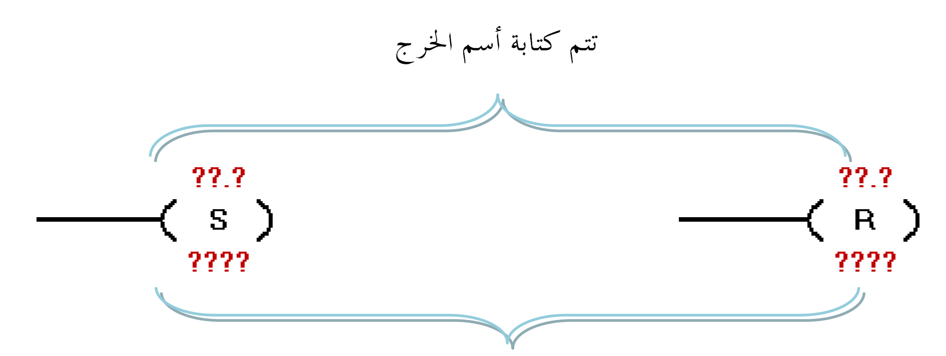
وهو يستخدم فى التشغيل, أي فى حالة إرسال إشارة إلى الـــ Set الخاص بأى output فأنة يعمل.

الـــ Reset:

??.?
—(R)
????

وهو يستخدم فى الفصل, أي فى حالة إرسال إشارة إلى الـــ Reset الخاص بأى output فأنه يفصل.

ملاحظة:



تتم كتابة عدد المخرجات المراد التعامل معها بداية من الخرج المكتوب فوق , وكأنهم متصلون بالتوازى

فمثلاً فى المثال التالى:

- بالضغط على I0.1 سوف يعمل الخرج Q0.0 و Q0.1 و Q0.2 معاً.

I0.1 Q0.0
S
3

- بالضغط على I0.0 سوف يفصل الخرج Q0.0 و Q0.1 و Q0.2 معاً.

I0.0 Q0.0
R
3

مقارنة بين الــ output العادى و الــ output الــ set/reset.

عدد النقاط	الإختلاف	الــ output العادى	الــ output الــ set/reset.
١	الشكل	??.? ( )	??.? (S) ???? ??.? (R) ????
٢	نقطة الحفظ	يمكن استخدامها.	لا يحتاج إلى نقطة الحفظ لأنة يحتفظ بحالة.
٣	التكرار	لا يمكن تكرار الــ output أكثر من مرة.	يجب تكرار الــ output مرة مع الــ set وأخرى مع الــ reset.
٤	عدد المخرجات	لا يمكن تشغيل أكثر من output ألا عن طريق التوصيل بالتوازى.	يمكن تشغيل أكثر من output عن طريق كتابة العدد تحت الــ Set أو الــ Reset.
٥	المفاتيح	فى حالة الفصل يستخدم مفتاح مغلق.	فى حالة الفصل يستخدم مفتاح مفتوح.
٦	الأولوية	لا توجد أولوية لأن الــ output لا يتكرر.	توجد أولوية لأن الــ output يتكرر.

مثال:

✓ محرك يعمل من مكان واحد و يقف من مكان واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

شرح التمرين:

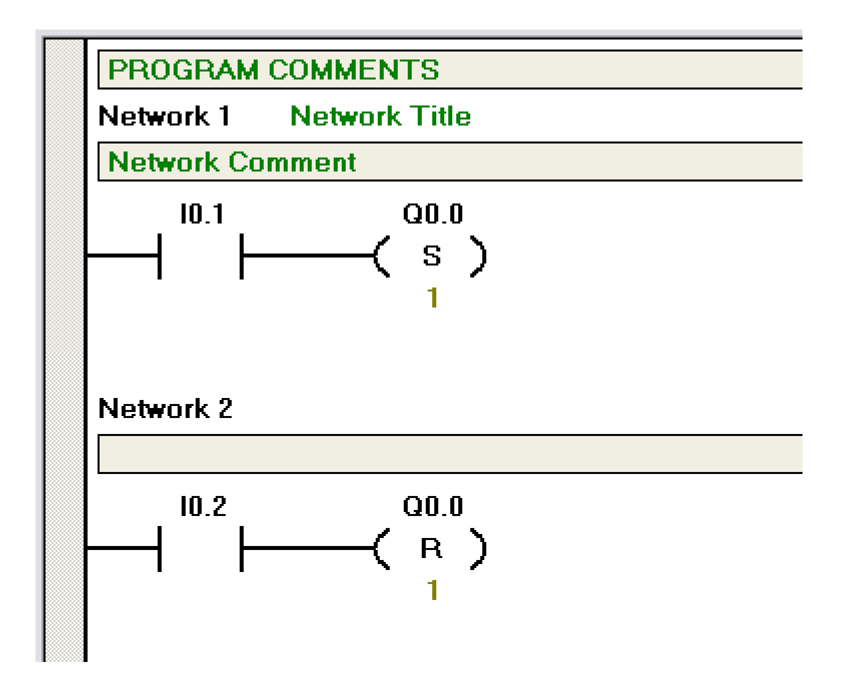
Network1

فى حالة الضغط على I0.1 تمر الإشارة إلى الــ Set فتعمل Q0.0 فقط بحيث أن رقم واحد المكتوب أسفل الــ Set يعنى أن Q0.0 سوف تعمل "لوحدها".

Network2

فى حالة الضغط على I0.2 تمر الإشارة إلى الــ Reset فتفصل Q0.0 فقط بحيث أن رقم واحد المكتوب أسفل الــ Reset يعنى أن Q0.0 سوف تفصل "لوحدها".

البرنامج:


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
Q0.0
S
1
Network 2
I0.2
Q0.0
R
1

مثال أخر:

- ✓ محركين يعملان بنفس الشروط, بحيث بالضغط على أحدى المفتاحين يعمل المحركين معاً و بالضغط على أحدى المفتاحين الأخريين يقف المحركين معاً.

المدخلات و المخرجات المستخدمة:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
٣	n.o.	I0.3/S3
٤	n.o.	I0.4/S4
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.1/K1M
٢	كونتاكتور	Q0.2/K2M

شرح التمرين:

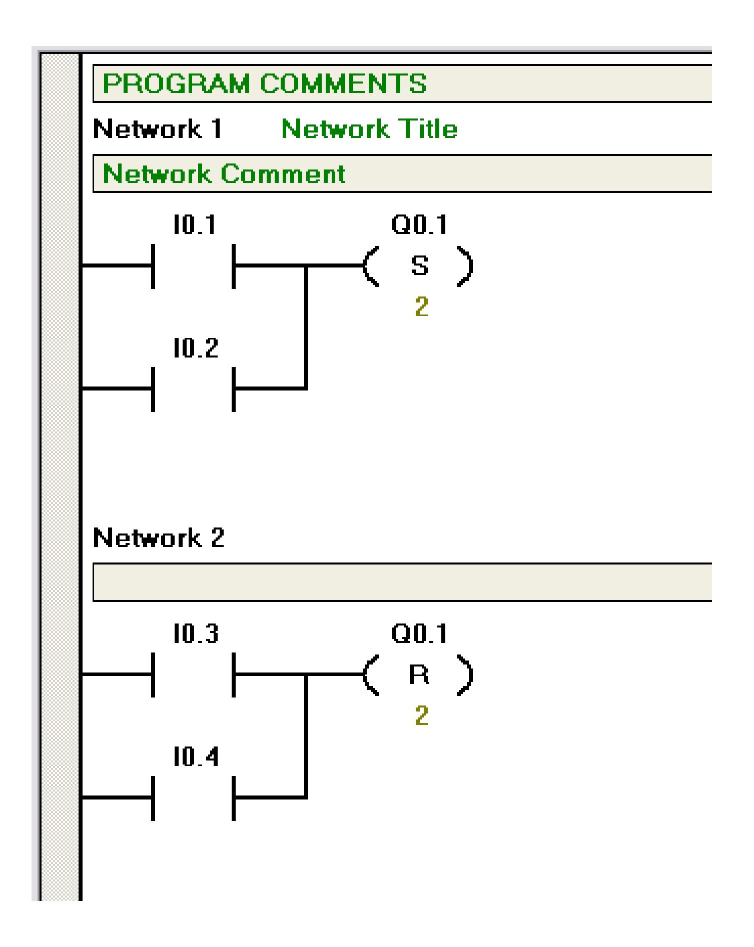
Network1

فى حالة الضغط على I0.1 أو I0.2 تمر الإشارة إلى الــ Set فتعمل Q0.1 و Q0.2 فقط بحيث أن رقم أثنين المكتوب أسفل الــ Set يعنى أن Q0.1 سوف تعمل ومعها الخرج التالى لها.

Network2

فى حالة الضغط على I0.3 و I0.4 تمر الإشارة إلى الــ Reset فتفصل Q0.1 و Q0.2 فقط بحيث أن رقم أثنين المكتوب أسفل الــ Reset يعنى أن Q0.1 سوف تفصل هى و الخرج التالى لها.

البرنامج:



ملاحظة:

تم توصيل المفتاحين المستخدمين في الفصل (Reset) على التوازى لأن الـــ Reset يعمل عندما يستقبل الإشارة من أحدى أو كلتا المفتاحين.

مثال أخر باستخدام الـــ Set/Reset:

✓ محرك يعمل فى اتجاهين باستخدام مفتاحين للتحكم باتجاه الحركة و مفتاح واحد للإيقاف.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
٣	n.o.	I0.3/S3
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.1/K1M
٢	كونتاكتور	Q0.2/K2M

شرح التمرين:

Network1

فى حالة الضغط على I0.1 تمر الإشارة إلى الـــ Set فتعمل Q0.1 فقط لأن رقم واحد المكتوب أسفل الـــ Set يعنى أن Q0.1 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.1 أثناء عمل Q0.2

Network2

فى حالة الضغط على I0.3 تمر الإشارة إلى الـــ Reset فتقف Q0.1 فقط لأن رقم واحد المكتوب أسفل الـــ Reset يعنى أنQ0.1 سوف تقف وحدها.

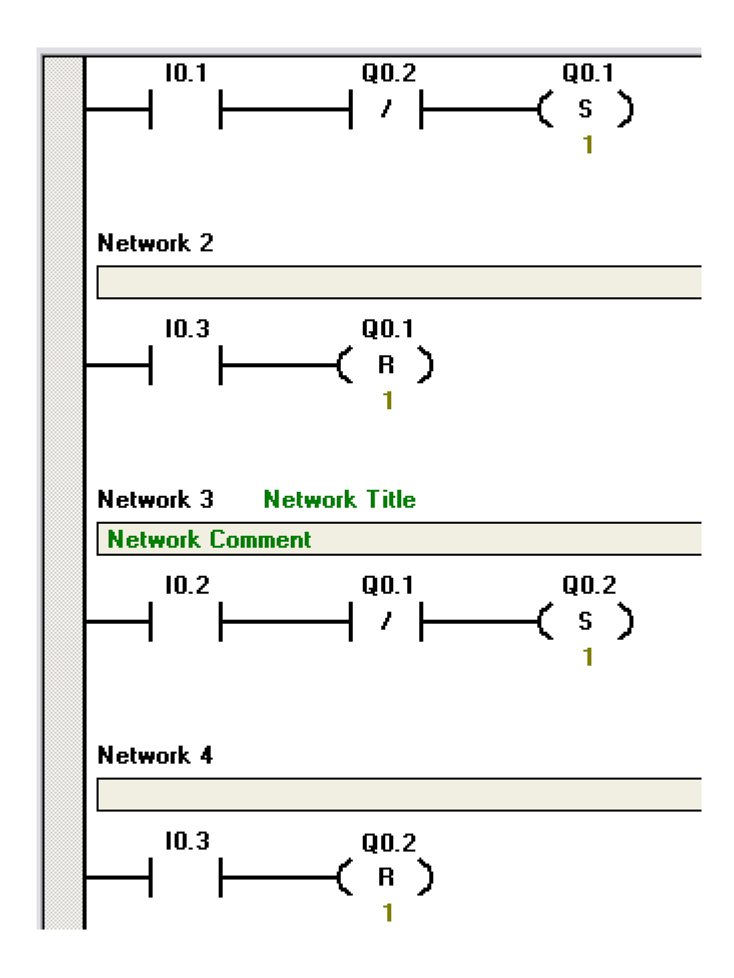
Network3

في حالة الضغط على I0.2 تمر الإشارة إلى الــ Set فتعمل Q0.2 فقط لأن رقم واحد المكتوب أسفل الــ Set يعني أن Q0.2 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.2 أثناء عمل Q0.1

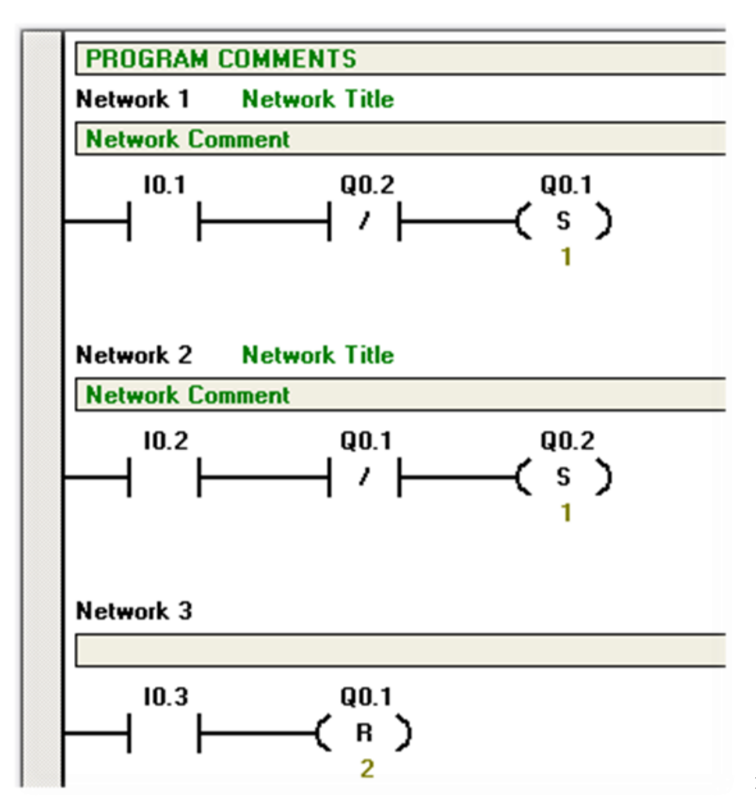
Network4

في حالة الضغط على I0.3 تمر الإشارة إلى الــ Reset فتقف Q0.2 فقط لأن رقم واحد المكتوب أسفل الــ Reset يعني أن Q0.2 سوف تقف وحدها.

البرنامج:



طريقة أخرى



شرح التمرين:

:Network1

فى حالة الضغط على I0.1 تمر الإشارة إلى الـــ Set فتعمل Q0.1 فقط لأن رقم واحد المكتوب أسفل الـــ Set يعنى أن Q0.1 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.1 أثناء عمل Q0.2

:Network2

فى حالة الضغط على I0.2 تمر الإشارة إلى الـــ Set فتعمل Q0.2 فقط لأن رقم واحد المكتوب أسفل الـــ Set يعنى أن Q0.2 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.2 أثناء عمل Q0.1

:Network3

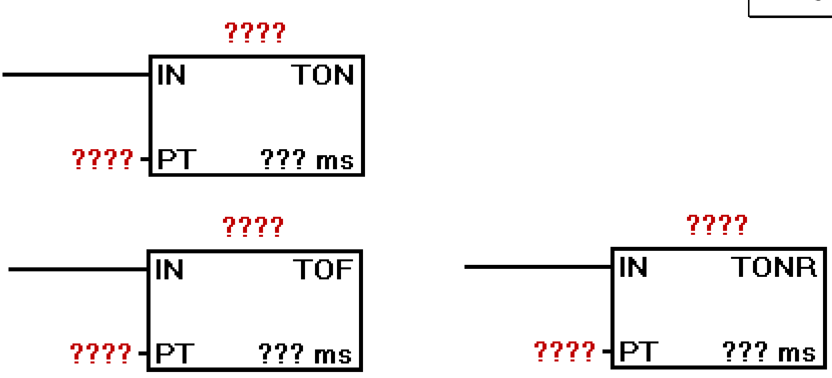
فى حالة الضغط على I0.3 تمر الإشارة إلى الـــ Reset الخاص بـــ Q0.1 و Q0.2 لأن رقم أثنين المكتوب أسفل الـــ Reset يعنى أن سواء كان يعمل المحرك يميناً أو شمالاً سوف يقف.

الباب الخامس

المؤقتات الزمنية

- أنـــــــــــــــــــــواع المؤقتات داخل الـــ **PLC**.
- مؤقـــــــــــت التشغيل المتأخـــــــــ**TON**ـــــــــر.
- مؤقـــــــــــت الفصل المتأخـــــــــ**TOF**ــــــــــر.
- مؤقـــــــت التشغيل المتـــأخــــر المتـــ**TONR**ــــــد.
- كـــيـــفـــيـــة تغير دقة المؤقت الزمنـــــــــــــــــى.
- مسميــــــــــات المؤقتات الزمنية فى الـــ **CPU 224**.
- مسميــــــــــات المؤقتات الزمنية فى الـــ **CPU 214**.
- رســـــــــــــــــــم تخطيطى للمؤقتات الزمنية الثلاثة.
- تماريـــن تطبيقية على المؤقتـــــــــــات الزمنيـــــــــة.

المؤقتات الزمنية:



المؤقتات الزمنية الأكثر استخداماً فى جهاز الـــ **PLC** هى:

١- مؤقت التشغيل المتأخر TON(Timer On Delay).

٢- مؤقت الفصل المتأخر TOF(Timer Off Delay).

٣- مؤقت التشغيل المتأخر الممتد TONR(Retentive Timer On Delay).

المسميات الخاصة بالمؤقتات الزمنية الـــ TON/TOF:

م	النوع	الوقت	الدقة	أقصى زمن	أسم المؤقتات
١	TON/TOF	١ثانية = ١٠٠٠	١ms	٣٢ ,٧٦٧ sec.	T32 ; T96
٢	TON/TOF	١ثانية = ١٠٠	١٠ms	٣٢٧ ,٦٧ sec.	T33 → T36 ; T97 → T100
٣	TON/TOF	١ثانية = ١٠	١٠٠ms	٣٢٧٦ ,٧ sec.	T37 → T63 ; T101 → T255

المسميات الخاصة بالمؤقتات الزمنية الــ TONR:

م	النوع	الوقت	الدقة	أقصى زمن	أسم المؤقتات
١	TONR	١ثانية = ١٠٠٠	ms١	٣٢ ,٧٦٧	T0 ; T64
٢	TONR	١ثانية = ١٠٠	ms١٠	٣٢٧ ,٦٧	T1 → T4 ; T65 → T68
٣	TONR	١ثانية = ١٠	ms١٠٠	٣٢٧٦ ,٧	T5 → T31 ; T69 → T95

خصائص المؤقتات الزمنية:

أولاً:Integer ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية يجب أن تكون أرقام صحيحة فقط وليست أرقام عشرية.

ثانياً:Word ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية تكتب على ذاكرة مكونة من 16bits وليس أكثر ولا أقل.

ثالثاً:Signed ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية يمكن أن تكون أرقام موجبة أو أرقام سالبة مع العلم أنه لا يوجد زمن بالسلب ولكن سنوضح فيما بعد السبب.

ملاحظة:

١- أقصى زمن للمؤقت الزمنى هو أقصى رقم موجب يمكن أن يكتب على ذاكرة "word" وهو من +٣٢٧٦٧ إلى 32768-.

٢ - لا يوجد مؤقت زمنى TOF فى الــ CPU214.

٣- لاختيار الدقة المطلوبة يكفى كتابة أسم المؤقت الزمنى حسب الجدول السابق.

توضيح لمعرفة دقة المؤقت الزمنى الخاص بالثلاث مؤقتات الزمنية.

م	توضيح	الشكل
١	مؤقت زمني TON أسمة T32 بزمن مسبق يساوى ٥ ثواني و بدقة ١ms.	T32 IN TON 5000 PT 1 ms
٢	مؤقت زمني TON أسمة T102 بزمن مسبق يساوى ٥ ثواني و بدقة ١٠٠ms.	T102 IN TON 50 PT 100 ms
٣	مؤقت زمني TOF أسمة T97 بزمن مسبق يساوى ٢ ثانية و بدقة ١٠ ms.	T97 IN TOF 200 PT 10 ms
٤	مؤقت زمني TOF أسمة T255 بزمن مسبق يساوى ٢ ثانية و بدقة ١٠٠ ms.	T255 IN TOF 20 PT 100 ms
٥	مؤقت زمني TONR أسمة T0 بزمن مسبق يساوى ١ ثانية و بدقة ١ms.	T0 IN TONR 1000 PT 1 ms
٦	مؤقت زمني TONR أسمة T5 بزمن مسبق يساوى ١ ثانية و بدقة ١٠٠ms.	T5 IN TONR 10 PT 100 ms

شرح كيفية عمل المؤقتات:

م	الأسم	الشرح	الشكل
١	TON مؤقت التشغيل المتأخر	يقوم بتغير وضعية الأقطاب الخاصة به بعد زمن من التشغيل أى انه يعتمد على الوقت المسبق. فى حاله انقطاع تغذية المؤقت الزمنى فأنه يعود إلى الصفر وبالضغط مرة أخرى يبدأ من البداية. يمكن تكرار نقاط المؤقت الزمنى دون حدود.	???? IN PT TON ??? ms ????
٢	TOF مؤقت الفصل المتأخر	يقوم بتغير وضعية الأقطاب الخاصة به عند تشغيله و بعد زمن من قطع الإشارة يبدأ المؤقت الزمنى بالعمل إلى أن يصل إلى صفر فترجع وضعية الأقطاب كما كانت فى البداية قبل أن يعمل المؤقت الزمنى. الوقت بين فصل المؤقت الزمنى و	???? IN PT TOF ??? ms ????

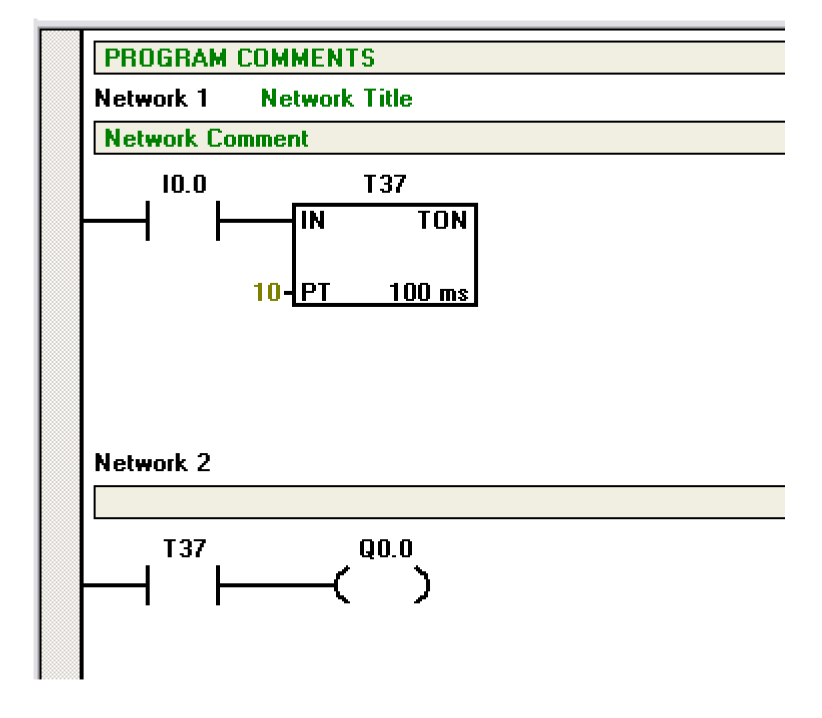
		رجوع وضعية الأقطاب إلى أصلها يعتمد على الوقت المسبق. يمكن تكرار نقاط المؤقت الزمنى دون حدود.	
٣	TONR مؤقت التشغيل المتأخر الممتد	يقوم بتغير وضعية الأقطاب الخاصة به بعد زمن من التشغيل أى انه يعتمد على الوقت المسبق. فى حاله انقطاع تغذية المؤقت الزمنى فأنه لا يعود إلى صفر وبالضغط مرة أخرى يستكمل العمل من نفس النقطة. يمكن تكرار نقاط المؤقت الزمنى دون حدود. لفصل المؤقت الزمنى نقوم بإرسال أشارة إلى الـــ reset الخاص بالمؤقت الزمنى.	???? IN TONR ???? PT ??? ms ??.? R ????

مثال TON :

محرك يعمل بعد زمن من الضغط على المفتاح.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T37
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

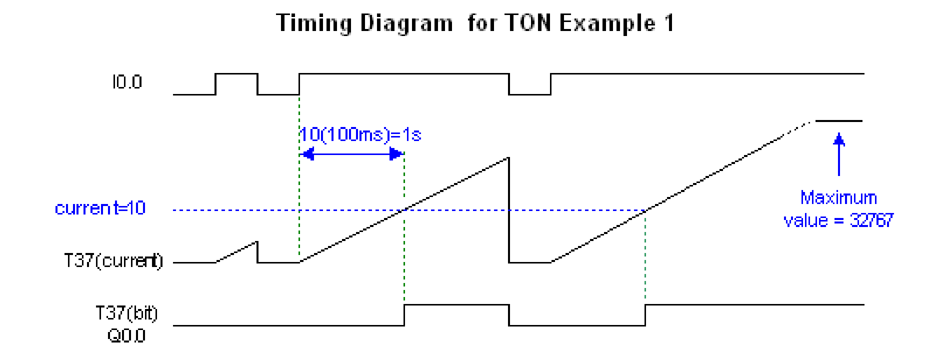
البرنامج:



الشرح:

بالضغط على المفتاح I0.0 يبدأ المؤقت الزمنى T37 فى العمل. عندما يصل إلى ثانية واحدة يعمل الخرج Q0.0 إلى أن يتم فصل المؤقت الزمنى T37 عن طريق فتح المفتاح I0.0

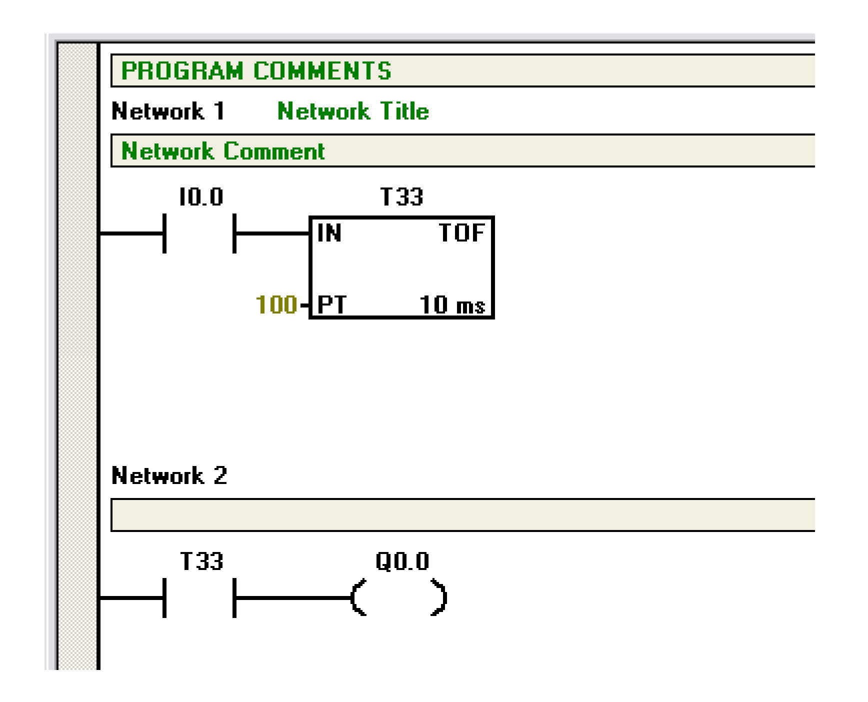
رسم تخطيطى:



مثال TOF :

محرك يعمل عند الضغط على المفتاح ولكن يفصل بعد زمن من فتح المفتاح.

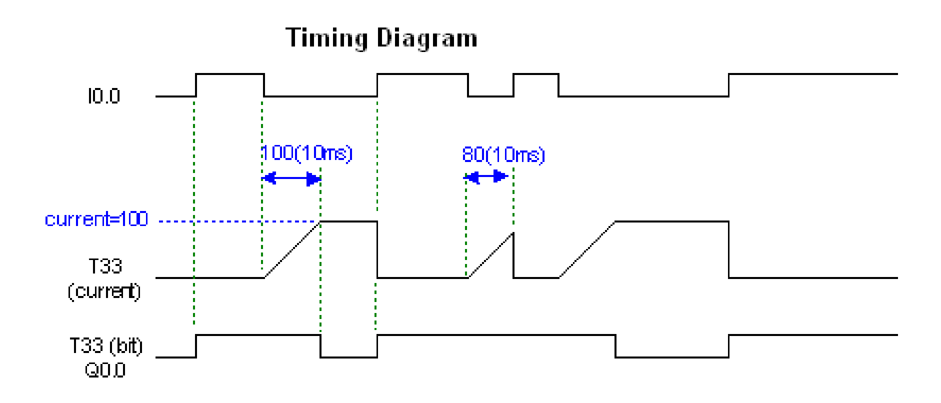
عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TOF	T33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M



الشرح:

بالضغط على المفتاح I0.0 يعمل المحرك ولكن عندما يفصل المفتاح يبدأ المؤقت الزمني T33 في العمل تنازلياً, بدايتاً من ثانية واحدة وحتى يصل المؤقت الزمني T33 إلى صفر يفصل الخرج Q0.0

رسم تخطيطي:



مثال TONR :

محرك يعمل بعد زمن من الضغط على المفتاح ولكن لا يفصل عند توقف المؤقت الزمنى بل عندما ترسل أشارة reset إلى الـــ Timer.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TONR	T37
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

الشرح:

بالضغط على المفتاح I0.0 يبدأ المؤقت الزمنى T1 فى العمل. عندما يصل إلى ثانية واحدة يعمل الخرج Q0.0 إلى أن يتم فصل المؤقت الزمنى T1 عن طريق إرسال أشارة إلى الـــ reset الخاص بالمؤقت الزمنى بواسطة المفتاح I0.1

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.0 T1

IN TONR

100 PT 10 ms

Network 2

T1 Q0.0

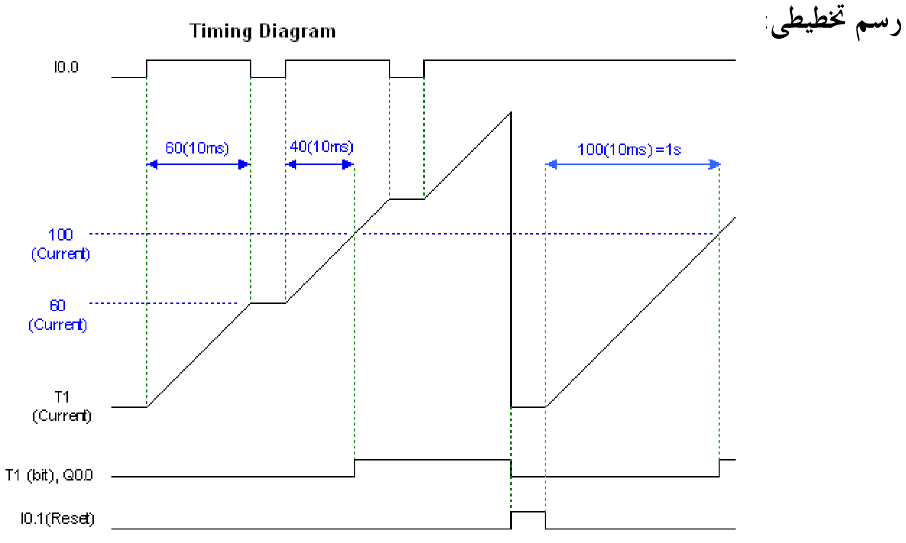
Network 3

I0.1 T1

R

1

رسم تخطيطي:



تمارين عملية على أنواع المؤقتات الزمنية..... Timers :

TON

١– قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من تشغيل المحرك الأول يدوياً.

TOF

٢– قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من إيقاف المحرك الأول يدوياً.

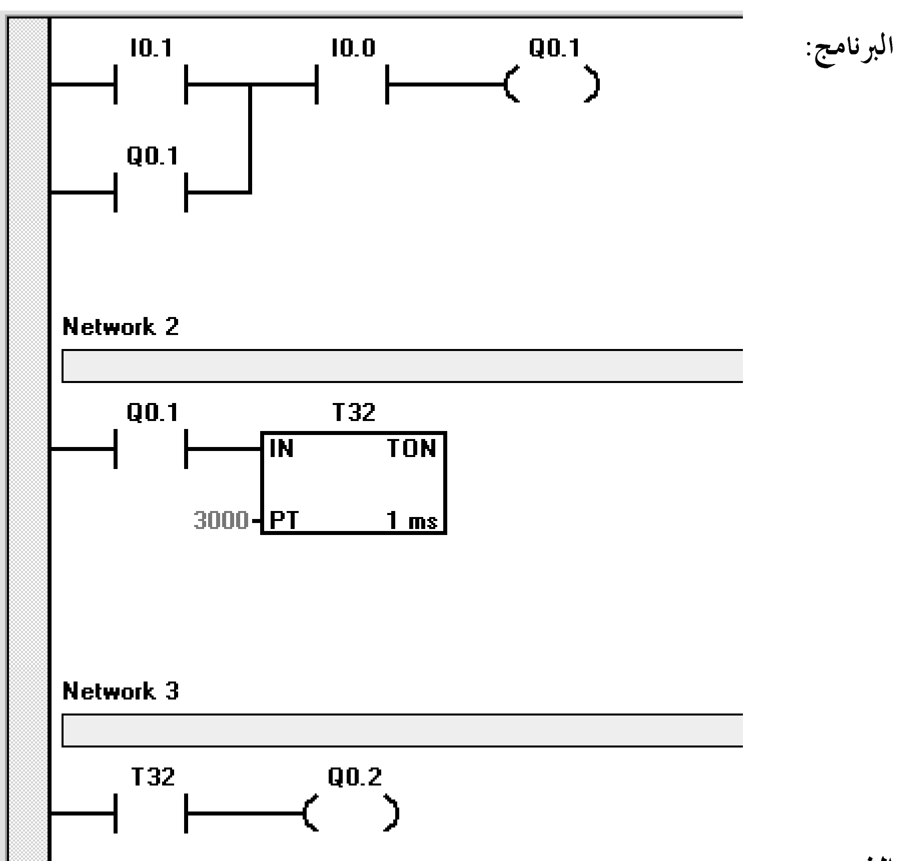
TONR

٣– بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من تشغيل المحرك الأول يدوياً مع مراعاة أنه فى حالة فصل الأول وتشغيله مرة أخرى يستكمل العد على الزمن القديم و بعد أن يعمل المحرك الثانى يتم فصل المؤقت الزمنى يدوياً للاستخدام مرة أخرى فيما بعد.

✓ **التمرين الأول باستخدام مؤقت زمنى TON:**

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M

البرنامج:



الشرح:

:Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q0.1 سوف تعمل فى الحال.

:Network2

فى نفس الدورة cycle التى ستعمل فيها Q0.1 سوف يعمل أيضاً المؤقت الزمنى T32.

:Network3

بعد مرور ٣ ثوانى وعند مرور الــ CPU على الــ Network3 سوف يعمل Q0.2

✓ **التمرين الثانى باستخدام مؤقت زمنى TOF**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I2.0/S1
٢	n.c.	I2.2/S2
٣	n.o.	I3.1/S3
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TOF	T100
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.5/K1M
٢	كونتكتور	Q4.4/K2M

الشرح:

:Network1

بالضغط على I3.1 و مع مراعاة أن المفتاح I2.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q4.4 سوف تعمل فى الحال.

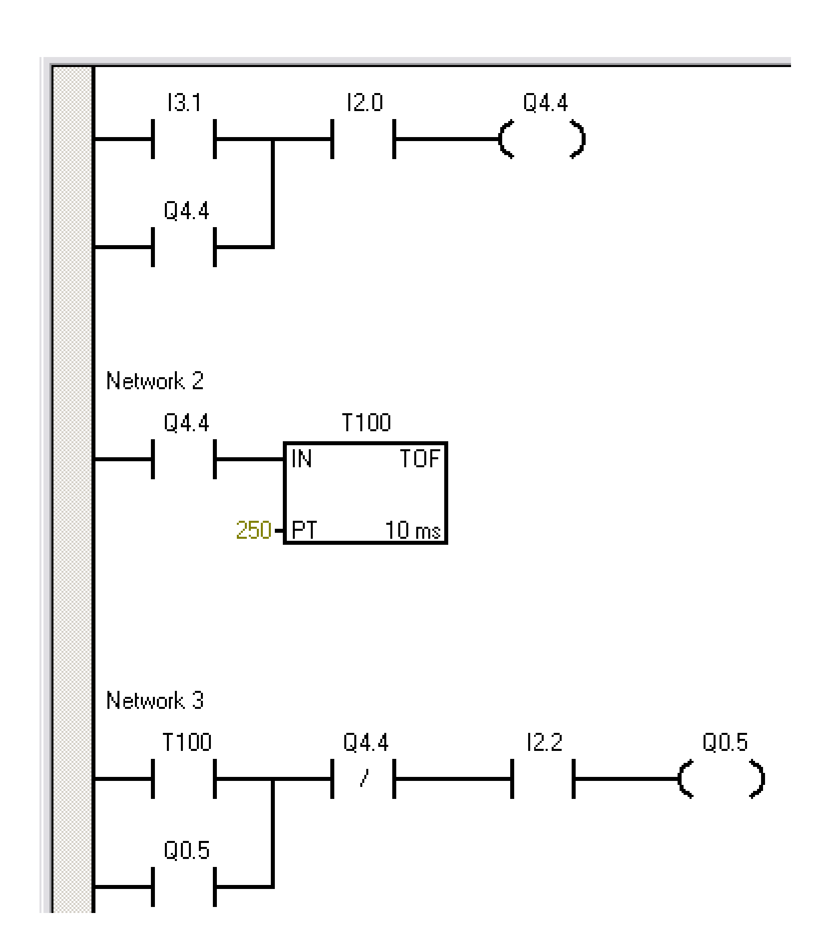
:Network2

فى نفس الدورة cycle التى ستعمل فيها Q4.4 سوف يعمل أيضاً المؤقت الزمنى T100 ويقوم بتغير وضعية الأقطاب الخاصة به ولكنه لن سيبدأ بالعد ألا عند انقطاع التغذية أى عند فصل Q4.4.

:Network3

عندما يتم فصل Q4.4 وبعد مرور ٥،٢ ثانية وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.2

البرنامج:


I3.1
I2.0
Q4.4
Q4.4
Network 2
Q4.4
T100
IN
TOF
250
PT
10 ms
Network 3
T100
Q4.4
I2.2
Q0.5
/
Q0.5

✓ **التمرين الثالث باستخدام مؤقت زمنى TONR**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.c.	I1.0/S2
٣	n.o.	I1.1/S3
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TONR	T0
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q2.2/K1M
٢	كونتكتور	Q2.4/K2M

الشرح:

Network1:

بالضغط على I1.1 و مع مراعاة أن المفتاح I1.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q2.2 سوف تعمل فى الحال.

Network2:

فى نفس الدورة cycle التى ستعمل فيها Q2.2 سوف يعمل أيضاً المؤقت الزمنى T0.

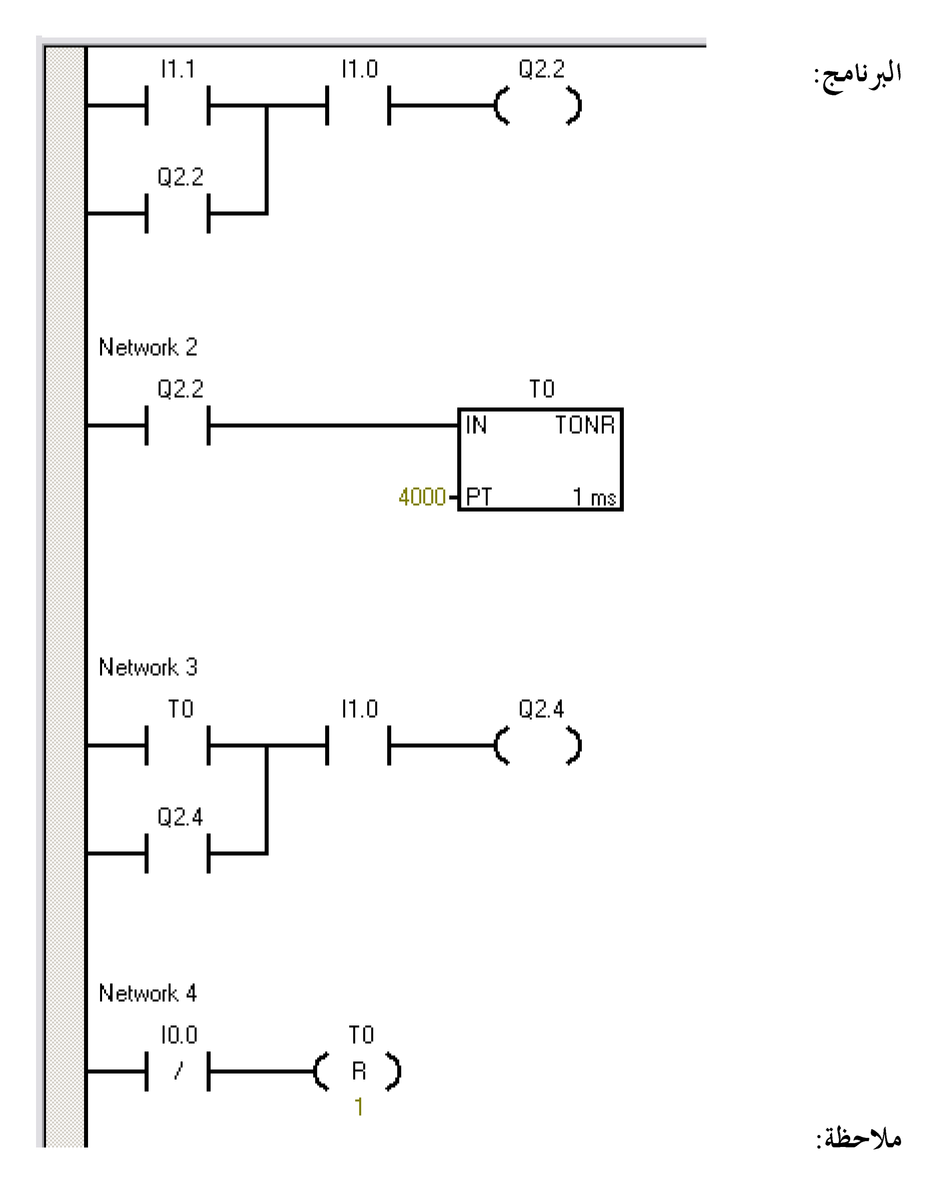
Network3:

بعد مرور ٤ ثوانى وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q2.4

Network4:

فى حاله فتح I0.0 بالخارج فأنه يغلق بالداخل فيعمل الـــ Reset فيرجع المؤقت الزمنى إلى صفر ليبدأ من جديد.

البرنامج:



ملاحظة:

- يفضل وضع نقطة Positive edge على التوالى بعد I0.0 لكى لا تستمر أشارة الـــ reset دائماً.
- في حاله فصل Q2.2 فأن المؤقت الزمنى T0 لا يرجع إلى صفر بل يتوقف وعند أعادة تشغيل Q2.2 مرة أخرى فأن المؤقت الزمنى T0 يستكمل من نفس القيمة التي قد توقف عندها قبل فصل Q2.2

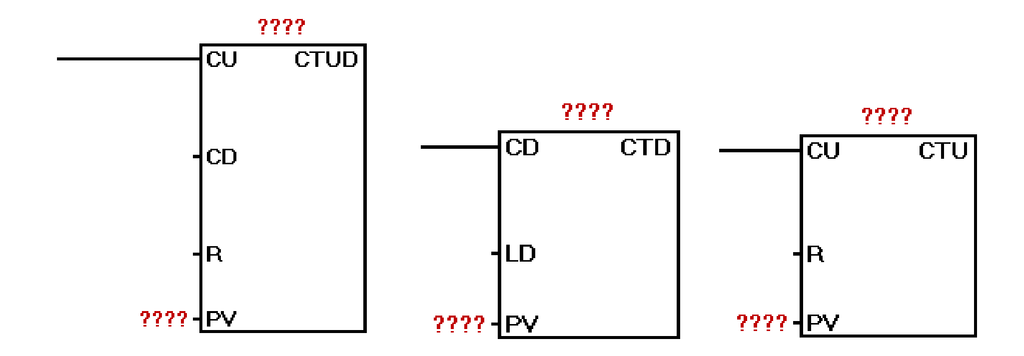
الباب السادس

العدادات

- أنـــــــــــــواع العـــدادات داخل الـــ PLC.
- عـــــــــــــ**CTU**ـــــــــــــــــــــداد تصاعدى.
- عـــــــــــــ**CTD**ـــــــــــــــــــداد تنازلـــى.
- عـــــ**CTUD**ــــــــداد تصاعدى و تنازلـــى.
- خـــــصـــــــائص العـــــــــــــــــــــداداث.
- مسميات العـــــــــــــــداداث فى الـــ CPU 224.
- مسميات العـــــــــــــــداداث فى الـــ CPU 214.
- رســـــــم تخطيطى للعـــــــــــــــداداث الثلاثة.
- تماريـــــــــــــــن تطبيقية على العـــــــــداداث.

العدادات:

تستخدم العدادات بمختلف أنوعها فى كثير من التمرين العملية حيث يمكن العداد أن يستخدم لمعرفة عدد القطع التى تم تصنعها, لمعرفة عدد السيارات داخل الجراج, لتعبئة مجموعة زجاجات فى كرتونه و للكثير من العمليات المختلفة التى سوف نتعرض لجزء منها فى ما بعد.



العدادات الأكثر أستخداماً فى جهاز الــ PLC هم:

١- عداد تصاعدى CTU (Counter Up).

٢- عداد تنازلى CTD (Counter Down).

٣- عداد تصاعدى و تنازلى CTUD (Counter Up and Down).

المسميات الخاصة بالعدادات الــ (CTU/CTD/CTUD(CPU 224:

م	النوع	العدات	الدقة	أقصى رقم موجب	أقصى رقم سالب	أسم العدادات
١	CTU	١عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255
٢	CTD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255
٣	CTUD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255

المسميات الخاصة بالعدادات الــ (CTU/CTD/CTUD(CPU 214:

م	النوع	العدات	الدقة	أقصى رقم موجب	أقصى رقم سالب	أسم العدادات
١	CTU	١عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C47
٢	CTD	لا يوجد	لا	يوجـــــد	لا	يوجـــــد
٣	CTUD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C48 → C127

خصائص العدادات الزمنية:

أولاً:**Integer** ← أى أن الأرقام المستخدمة مع العدادات يجب أن تكون أرقام صحيحة فقط.

ثانياً:**Word** ← أى أن الأرقام المستخدمة مع العدادات تكتب على ذاكرة مكونة من 16bits.

ثالثاً:**Signed** ← أى أن الأرقام المستخدمة مع العدادات يمكن أن تكون أرقام موجبة أو أرقام سالبة.

ملاحظة:

١ – أقصى رقم للعدادات هو أقصى رقم يمكن أن يكتب على ذاكرة word.

٢ – أمثلة عن شكل العدادات, فمثلاً:

م	توضيح	الشكل
١	عداد تصاعدى CTU أسمة C10 بقيمة مسبقة تساوى ٥ عدات.	C10 CU CTU R 5 - PV
٢	عداد تنازلى CTD أسمة C29 بقيمة مسبقة تساوى ١٥ عدة.	C29 CD CTD LD 15 - PV
٣	عداد تصاعدى و تنازلى CTUD أسمة C100 بقيمة مسبقة تساوى ٥٣ عدة.	C100 CU CTUD CD R 53 - PV

شرح كيفية تشغيل العدادات:

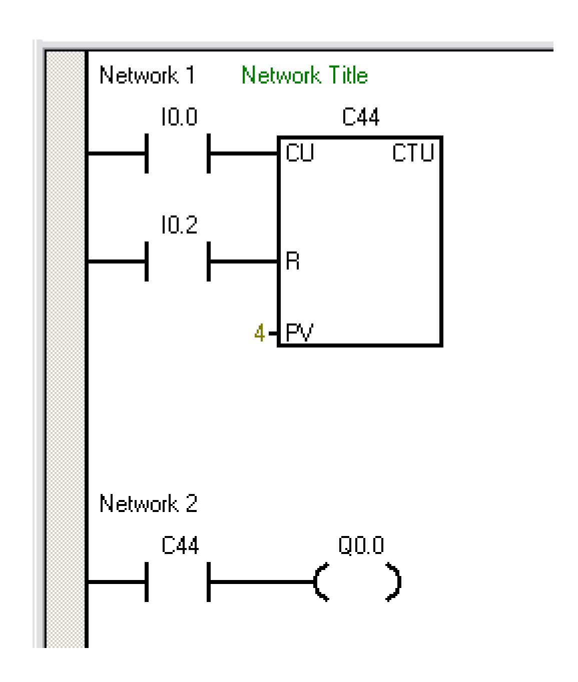
م	الأسم	الشرح	الشكل
١	**CTU** عداد تصاعدى	يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى القيمة المسبقة المحددة منذ البداية. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى صفر بل بإرسال أشارة إلى الـــ R فأنة يعود إلى صفر ليبدأ من البداية. يمكن تكرار نقاط العداد دون حدود. العداد CTU يقوم بالعد تصاعدياً بداية من صفر إلى أن يصل أو يتخطى القيمة المسبقة.	???? CU CTU R ???? PV
٢		يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى	

	CTD عداد تنازلى	صفر. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى القيمة المسبقة بل بإرسال إشارة إلى الـــ LD فأنة يعود إلى القيمة المسبقة ليبدأ من البداية. يمكن تكرار نقاط العداد دون حدود. العداد CTD يقوم بالعد تنازلياً بداية من القيمة المسبقة إلى أن يصل إلى صفر.	???? CD CTD LD ???? PV
٣	**CTUD** عداد تصاعدى و تنازلى	يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى القيمة المسبق المحددة منذ البداية. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى صفر بل بإرسال أشارة إلى الـــ R فأنة يعود إلى صفر ليبدأ من البداية	???? CU CTUD CD R ???? PV

		يمكن تكرار نقاط العداد دون حدود. العداد CTUD يقوم بالعد تصاعدياً و تنازلياً بداية من صفر إلى أن يصل أو يتخطى القيمة المسبقة.	

مثال CTU :

✓ محرك يعمل بعد أربع عدات.

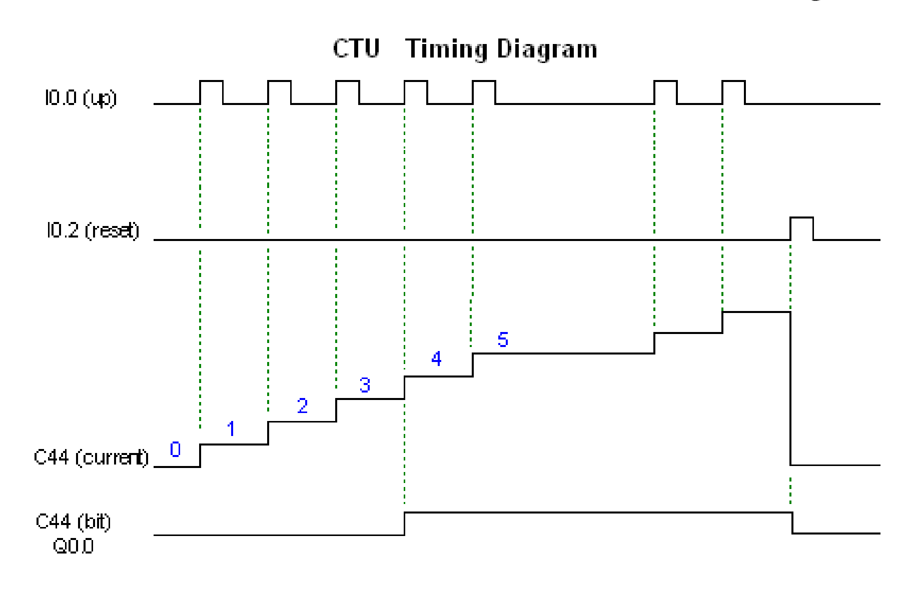


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.2/S1
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C44
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

الشرح:

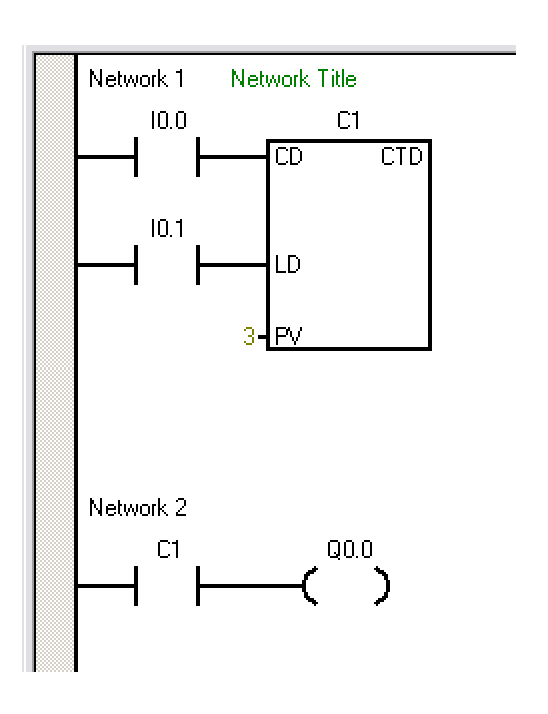
بالضغط على المفتاح I0.0 يبدأ العداد بالعد تصاعدياً وبعد أربع أشارات متفرقة يكون قد وصل العداد C44 إلى أربعة فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم تصفير العداد بواسطة المفتاح I0.2 للبدء من جديد.

رسم تخطيطى:



مثال CTD:

✓ محرك يعمل بعد ثلاث عدات.

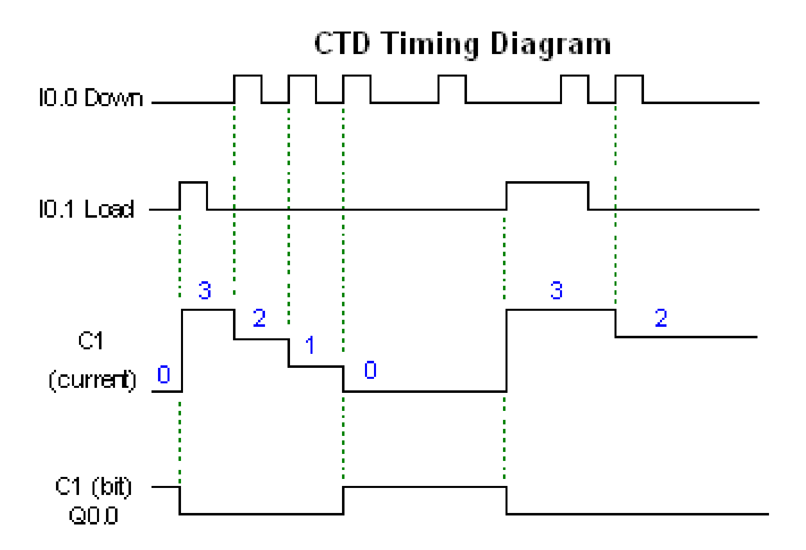


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.1/S1
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTD	C1
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

الشرح:

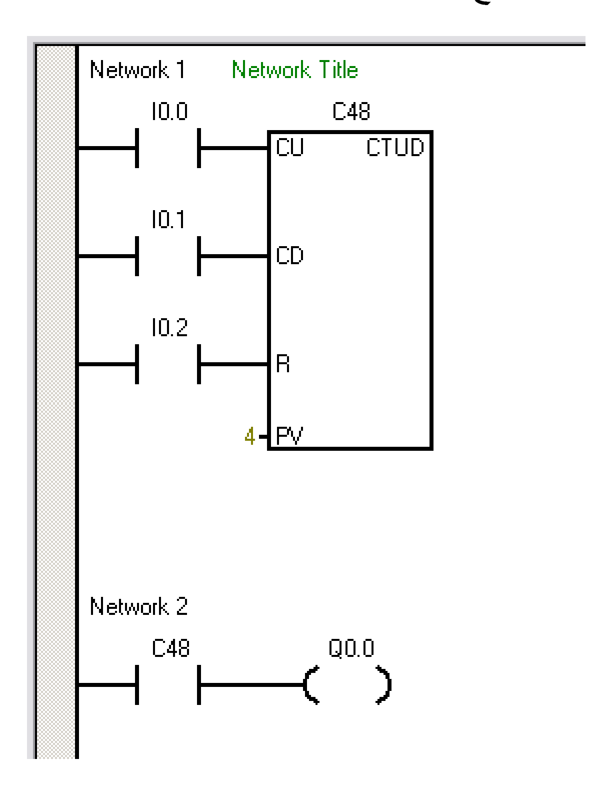
بالضغط على المفتاح I0.0 يبدأ العداد بالعد تنازلياً وبعد ثلاث أشارات متفرقة يكون قد وصل العداد C1 إلى صفر فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم أعادة الرقم إلى ثلاثة بواسطة المفتاح I0.1 للبدء من جديد.

رسم تختيطى:



مثال CTUD :

✓ محرك يعمل بعد أربع عدات.

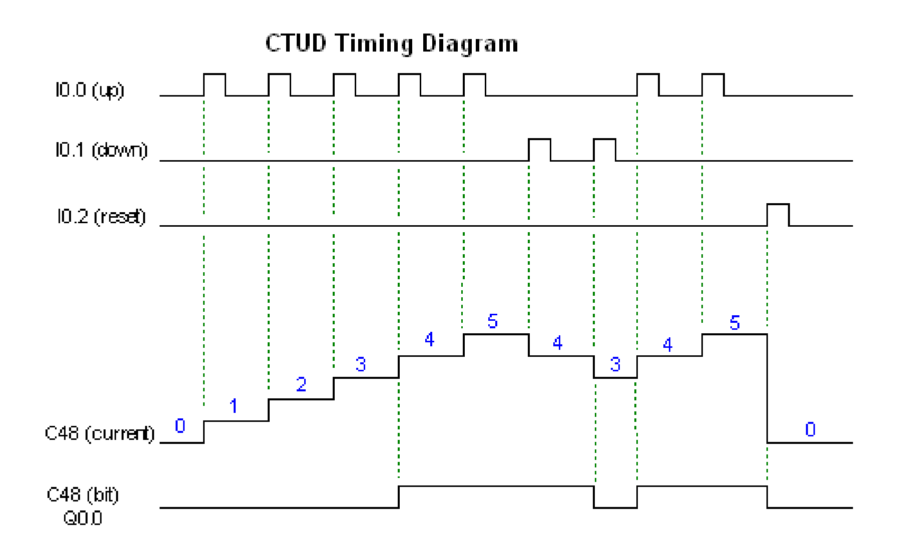


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.1/S1
عدد العدادات	نوع العدادات	أسم العدادات
١	CTUD	C48
عدد الخرج	نوع الخرج	أسم الخرج
١	كونتاكتور	Q0.0/K1M

الشرح:

بالضغط على المفتاح I0.0 يبدأ العداد بالعد تصاعدياً بينما بالضغط على المفتاح I0.1 يبدأ العداد بالعد تنازلياً وبعد وصول العداد إلى أربعة سواء بالضغط أربع مرات على I0.0 أربع مرات أو بأى طريقة أخرى بحيث يكون قد وصل العداد C44 إلى أربعة فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم تصفير العداد بواسطة المفتاح I0.2 للبدء من جديد.

رسم تخطيطى:



تمارين عملية على أنواع العدادات (counters):

CTU

١- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد أن يكون قد عمل المحرك الأول يدوياً خمس مرات متتالية بأستخدام عداد تصاعدى.

CTD

٢- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد أن يكون قد عمل المحرك الأول يدوياً خمس مرات متتالية باستخدام عداد تنازلى.

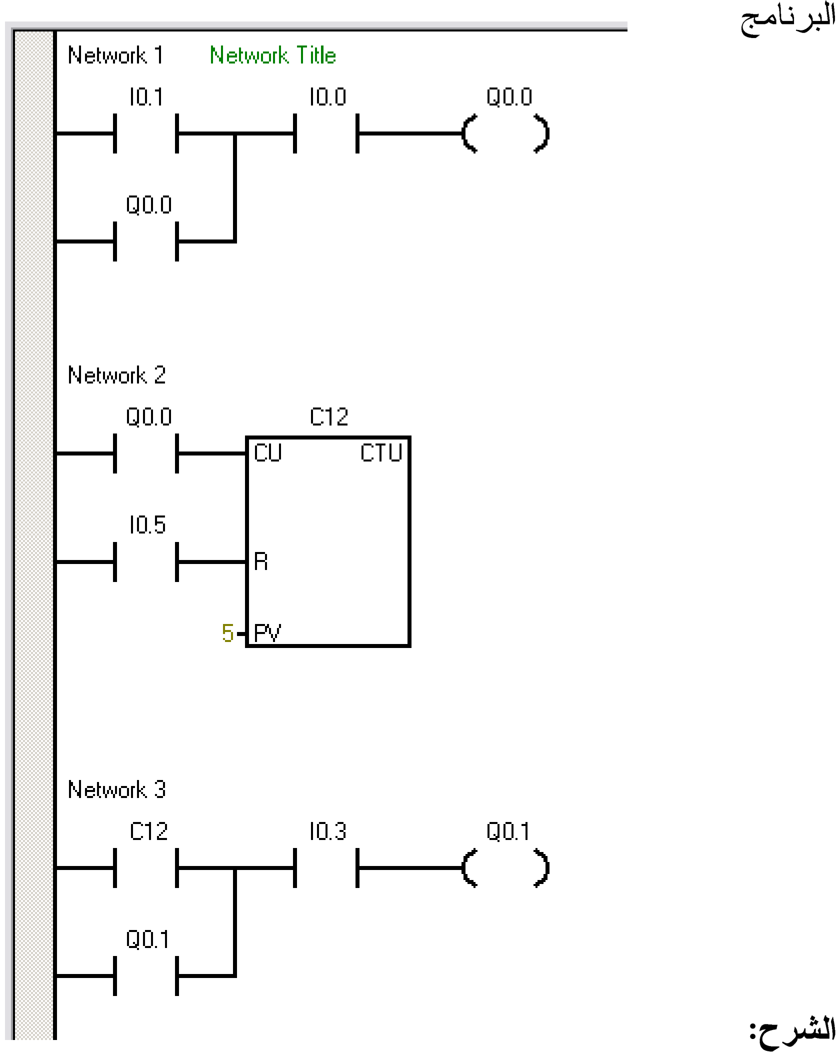
CTUD

٣- بتنفيز دائرة تحكم منطقية لخرجين (لمبات أشارة) بحيث أن الإشارة الخضراء تشير إلى أن الجراج فارغ بينما تشير الإشارة الحمراء إلى أن الجراج ممتلئ. علماً أن أقصى عدد سيارات داخل الجراج هو خمسة فأنة عندما يصل العدد إلى خمسة يجب أن تضاء الإشارة الحمراء لتشير أن عدد السيارات داخل الجراج أكتمل.

- ✓ **التمرين الأول باستخدام عداد تصاعدى CTU:**

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C12
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M
٢	كونتاكتور	Q0.1/K2M

البرنامج



الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة 110) فأن Q0.0 سوف تعمل فى الحال.

Network2

فى نفس الـــ cycle التى ستعمل فيها Q0.0 سوف يعمل العداد C12 ويبدأ بالعد تصاعدياً.

Network3

بعد تكرار هذه العملية خمس مرات وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.1

✓ **التمرين الثانى باستخدام عداد تصاعدى CTD:**

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTD	C33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M
٢	كونتاكتور	Q0.1/K2M

الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة 110) فأن Q0.0 سوف تعمل فى الحال.

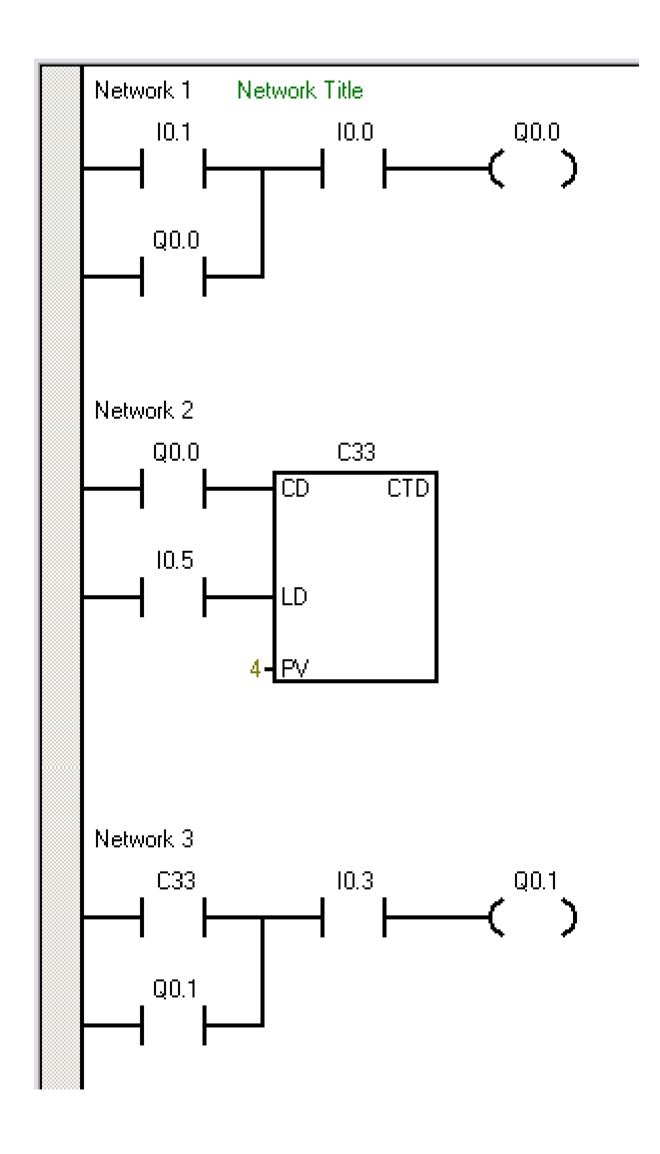
Network2

في نفس الـــ cycle التي ستعمل فيها Q0.0 سوف يعمل العداد C12 ويبدأ بالعد تنازلياً.

Network3

بعد تكرار هذه العملية أربع مرات وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.1

البرنامج



✓ **التمرين الثالث باستخدام عداد تصاعدى و تنازلى CTUD**:

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
٥	n.o.	I0.6/S5
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTUD	C200
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة حمرة	Q0.1/H1
٢	لمبة خضرة	Q0.2/H2

الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مفتوح بالخارج (أنظر صفحة 110) فأن العداد C200 سوف يبدأ بالعد تصاعدياً بينما بالضغط على I0.2 و مع مراعاة أن المفتاح I0.0 مفتوح بالخارج (أنظر صفحة 110) فأن العداد C200 سوف يبدأ بالعد تنازلياً.

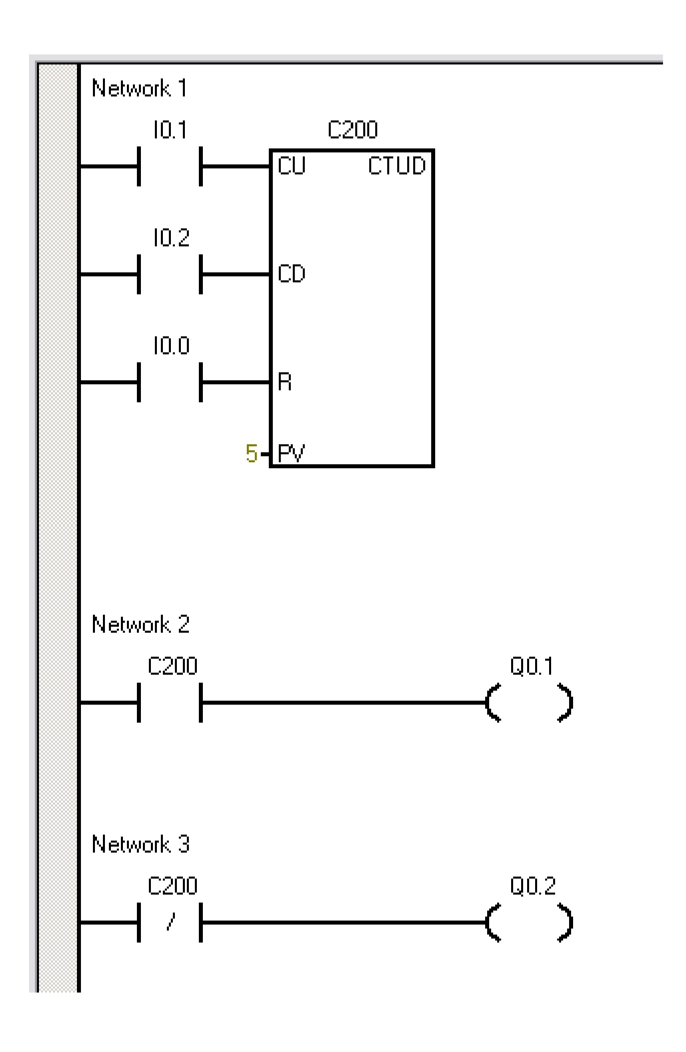
Network2

عندما يصل العداد إلى خمسة يقوم بغلق النقطة المفتوحة فتضئ الإشارة الحمراء لتشير أن الجراچ قد أمتلئ.

Network3

عندما يصل العداد إلى خمسة يقوم بفتح النقطة المغلقة فتضيء الإشارة الخضراء لتشير أن الجراج لم يعد ممتلئ.

البرنامج



الباب السابع

المتغيرات

- أنـــــــــــــــواع المتغيرات داخل الــ PLC.
- متغيـــــــــــــــــــــــــــــرات بحجم bit.
- متغيـــــــــــــــــــــــــــــرات بحجم byte.
- متغيـــــــــــــــــــــــــــــرات بحجم word.
- متغيـــــــــــــــــــــــــــــرات بحجم Dword.
- طريقة أستخدام المتغيـــــــــــــــــرات بحجم bit.
- مسميات المتغيـــــــــــــــــرات فى الــ PLC.
- تماريـــــــــــــــن تطبيقية على المتغيـــــــــــرات.

المتغيرات.........variables:

هى ذاكرة داخل الــ PLC تستخدم فى كتابة أى أرقام حتى تستخدم فى ما بعد سواء فى عمليات حسابية أو فى المقارنة إلخ...

تنقسم الذاكرة المتغيرة variables إلى:

bit - byte - word - Dword

١- V(BIT): هى أصغر وحدة للذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر أو واحد.

1

٢- V(BYTE): هى ذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 8 bits.

1	0	1	0	0	0	1	1

٣- V(WORD): هى ذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 2 bytes و 16 bits.

0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

٤- V(D.WORD): هى أكبر وحدة للذاكرة داخل جهاز الــ PLC وهى قد يحتوى على صفر و واحد وهو تتكون من 2 words و 4 bytes و 32 bits.

1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

هام:

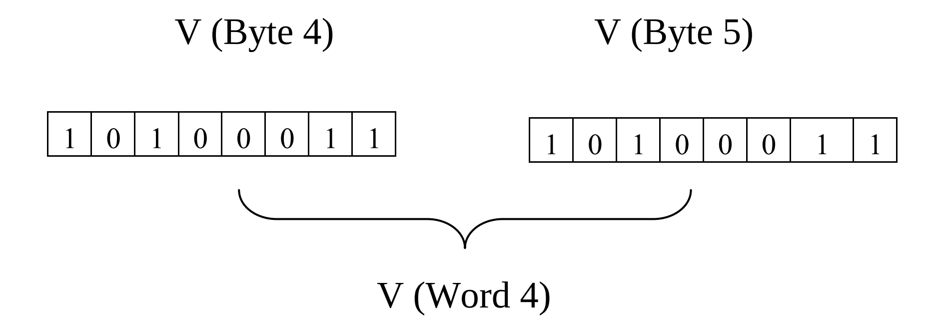
لكل وحدة من الذاكرة أسم و طريقة للكتابة:

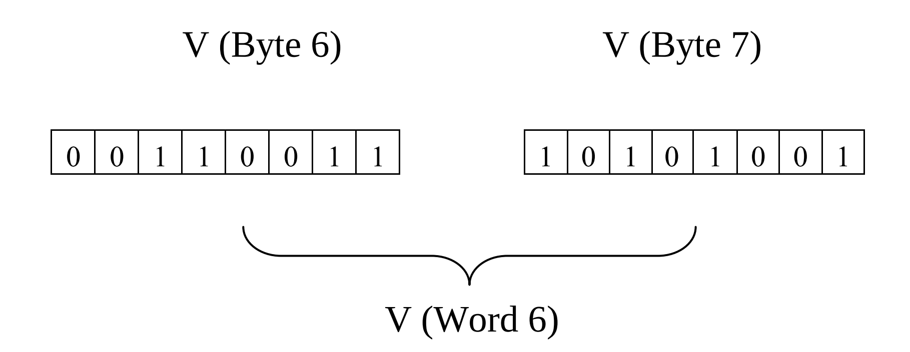
- ....VBIT 4, VBIT 3, VBIT 2, VBIT 1, VBIT0
- VBYTE 0, VBYTE 1, VBYTE 2, VBYTE 3, ...
- VWORD 0, VWORD 2, VWORD 4, VWORD 6, ...
- VD.WORD 0, VD.WORD 4, VD.WORD 8, VD.WORD 12

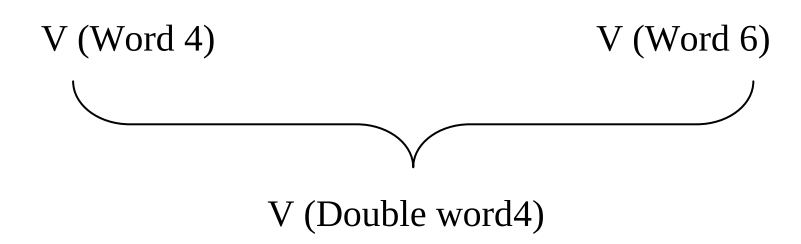
ملاحظة:

من المهم جداً مرعاه أن فى حالة كتابة معلومات بواسطة الـــ bits يجب بدء الكتابة من اليمين إلى اليسار أما فى حالة كتابة معلومات بواسطة الـــ bytes أو الـــ words أو الـــ D.words يجب بدء الكتابة من اليسار إلى اليمين وكذالك أيضاً فى حالة قراءة البيانات.

شرح مفصل:







لفهم طريقة القراءة أنظر صفحة 64

الأستخدامات:

يستخدم الــ **V**bit مثل الريليه تماماً مثلاً:

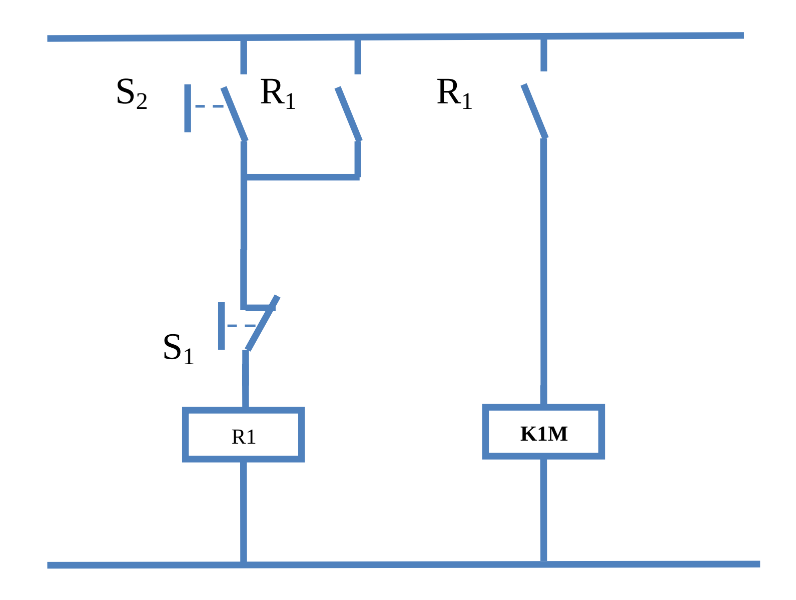
- V0.3 , V0.2 , V0.1 , V0.0

بينما يستخدم الــ **V**Byte و **V**Word و **VD**.word كذاكرة لتخزين البيانات مثلاً:

- VB2 , VB1 , VB0
- VW4 , VW2 , VW0
- VD8 , VD4 , VD0

✓ مثال باستخدام الـ Vbit مثل الـريليه:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المتغيرات	نوع المتغيرات	أسم المتغيرات
١	bit	V0.0
عدد الخرج	نوع الخرج	أسم الخرج
١	كونتكتور	Q0.2/K1M



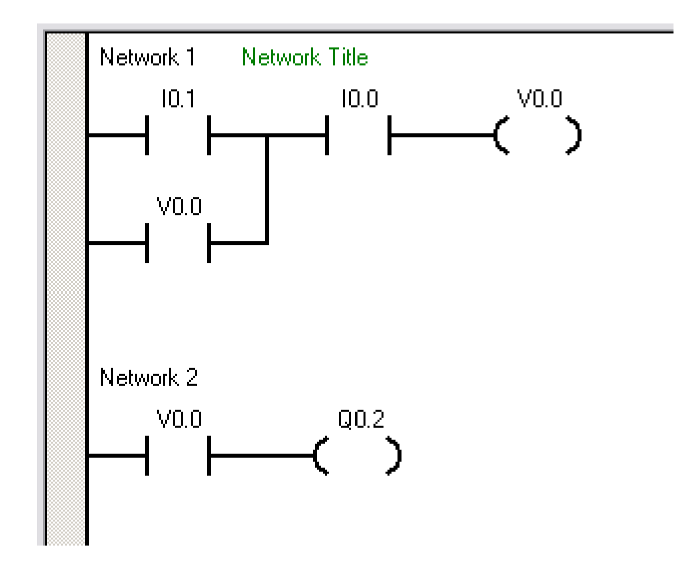
Network1:

بالضغط على I0.1 يعمل الـــ V0.0 مثل الـــ MARKER أو الـــ RELAY.

Network2:

عندما يعمل الـــ V0.0 يعمل أيضاً المحرك Q0.2

البرنامج:



- مثال بأستخدام الـــ Vword:

يمكن استخدام المتغيرات مع المؤقتات الزمنية كما بالتمرين التالى:

- ✓ عندما يصل المؤقت الزمنى إلى وقت معين يجب أن يقوم بفصل لمبة و أضاءه الأخرى بشرط أن يكون الزمن قابل للتغيير.

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.o.	I0.0/S1
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	word	VW0
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M
٢	كونتكتور	Q0.1/K2M

الشرح:

:Network1

بالضغط على I0.0 يبدأ المؤقت الزمنى T200 بالعمل.

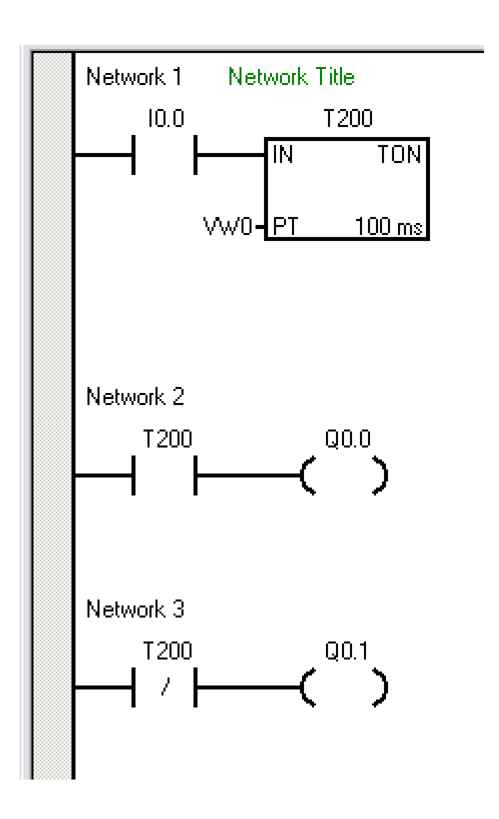
:Network2

عندما يصل المؤقت الزمنى إلى الوقت المحدد بواسطة المتغيرات يقوم بتشغيل الخرج Q0.0

:Network3

عندما يصل المؤقت الزمنى إلى الوقت المحدد بواسطة المتغيرات يقوم بفصل الخرج Q0.1

البرنامج:



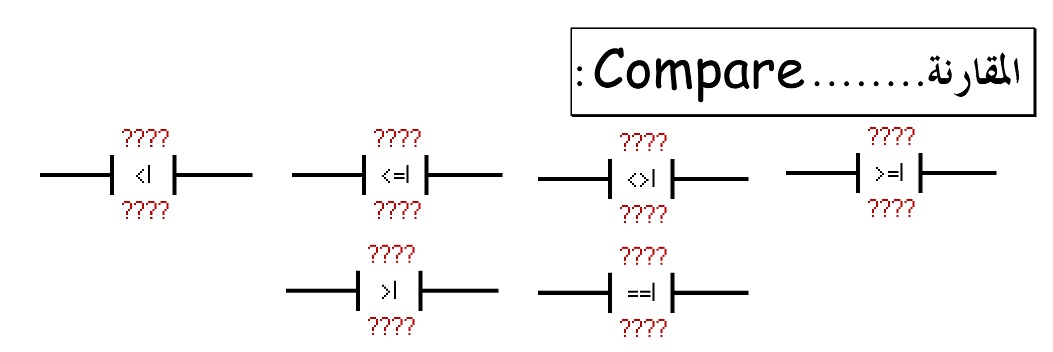
ملاحظة:

- فى البداية تكون قيمة المتغيرات صفر وهذا يسبب مشكلة كبيرة فى هذا التمرين لأنه فى هذا التمرين سيقوم المؤقت الزمنى بتغير النقاط الخاصة به عندما يصل إلى صفر أى فى نفس اللحظة التى سيغلق فيها المفتاح I0.0 فسيبدو أنه يعمل مثل الريليه تماماً.
- لحل هذه المشكلة يتم استخدام صفحة المتغيرات لتحديد قيمة مسبقة لأى من المتغيرات المستخدمة فى البرنامج.
- القيمة المسبقة التى تحدد بواسطة صفحة المتغيرات قابلة للتغير أثناء تنفيذ البرنامج كما سنلاحظ فيما بعد "الجزء الثانى".

الباب الثامن

المقارنة

- أنواع مفاتيح المقـــــــــــارنة داخل الـــ PLC.
- مفاتيح المقـــــــــــارنة داخل الـــ CPU 214.
- مفاتيح المقـــــــــــارنة داخل الـــ CPU 224.
- مفاتيح المقـــــــــــــــــــــارنة نوع Byte.
- مفاتيح المقـــــــــــــــــــــارنة نوع Word.
- مفاتيح المقـــــــــــــــــــــارنة نوع Dword.
- مسميات مفاتيح المقـــــــــــــــــــــارنة.
- تماريـــــــــن تطبيقية بأستخدام مفاتيح المقـــــارنة.



كل مفتاح من مفاتيح المقارنة هو عبارة عن معادلة بحيث أنة عندما تتحقق هذه المعادلة يصبح المفتاح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.

كل مفاتيح المقارنة توجد فى الــ CPU224 ولكن ليست جميعها توجد فى الــ CPU214.

م	CPU 214	CPU 224	مفتاح المقارنة
١	√	√	???? ==| ????
٢	---------	√	???? <>| ????
٣	√	√	???? >=| ????
٤	---------	√	???? >| ????
٥	√	√	???? <=| ????
٦	---------	√	???? <| ????

كما أنه سيتم شرح كيفية قراءة المعادلة الخاصة بكل مفتاح من مفاتيح المقارنة على حدا فى الجدول التالى.

م	الأسم	الشرح	الشكل
١	**Equal to** **يساوى**	عندما تتساوى القيمة المكتوبة فوق المفتاح مع القيمة المكتوبة أسفل المفتاح و العكس صحيح يصبح المفتاح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? ==I ????
٢	**Greater than or equal** **أكبر من أو يساوى**	عندما تصبح القيمة المكتوبة فوق المفتاح أكبر من أو تساوى القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أصغر من أو تساوى القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? >=I ????
٣		عندما تصبح القيمة المكتوبة فوق المفتاح أصغر من	

	Less than or equal **أصغر من أو يساوى**	أو تساوى القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أكبر من أو تساوى القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <=I ????
٤	**Not equal to** **لا يساوى**	عندما لا تتساوى القيمة المكتوبة فوق المفتاح مع القيمة المكتوبة أسفل المفتاح و العكس صحيح يصبح المفتاح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <>I ????
٥	**Greater than** **أكبر من**	عندما تصبح القيمة المكتوبة فوق المفتاح أكبر من القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أصغر من القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? >I ????

٦	**Less than** **أصغر من**	عندما تصبح القيمة المكتوبة فوق المفتاح أصغر من القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أكبر من القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <I ????

أنواع المقارنات الموجودة بجهاز الــ PLC:

	م	الأنواع	توضيح	الشكل
أنواع المقارنات	١	Byte	يستخدم لمقارنة Byte مع Byte ولهذا أكبر رقم يمكن كتابته هو ٢٥٥	???? ==B ????
	٢	Word	يستخدم لمقارنة Word مع Word ولهذا أكبر رقم موجب يمكن كتابته هو +٣٢٧٦٧ بينما أكبر رقم سالب هو -٣٢٧٦٨	???? ==I ????
	٣	D.Word	يستخدم لمقارنة D.Word مع D.Word ولهذا أكبر رقم يمكن كتابته هو ٤٢٩٤٩٦٧٢٩٥	???? ==D ????

أنواع المقارنات	٤	Real	يستخدم لمقارنة D.Word مع D.Word وهذا لأن الأرقام التي هي بالعلامة العشرية تكتب فقط على D.Word	???? ==R ????
	٥	String	يستخدم لمقارنة Byte مع Byte وهذا لأن الحرف الواحد يكتب فقط على Byte.	???? ==S ????

كل ما يمكن كتابته فوق أو تحت مفتاح المقارنة:

م	النوع	المكان	الرموز المستخدمة
١	???? ==B ????	مسميات فوق المفتاح	ثوابت , IB , QB , MB , VB , AC , SMB
		مسميات أسفل المفتاح	ثوابت , IB , QB , MB , VB , AC , SMB
٢	???? ==I ????	مسميات فوق المفتاح	ثوابت , IW , QW , MW , VW , C , T , AC , SMW
		مسميات أسفل المفتاح	ثوابت , IW , QW , MW , VW , AIW , C , T , AC , SMW , AQW

٣	???? ==D ????	مسميات فوق المفتاح	ثوابت , ID , QD , MD , VD , AC , SMD
		مسميات أسفل المفتاح	ثوابت , ID , QD , MD , VD , AC , SMD
٤	???? ==R ????	مسميات فوق المفتاح	ثوابت , ID , QD , MD , VD , AC , SMD
		مسميات أسفل المفتاح	ثوابت , ID , QD , MD , VD , AC , SMD
٥	???? ==S ????	مسميات فوق المفتاح	ثوابت , IB , QB , MB , VB , AC , SMB
		مسميات أسفل المفتاح	ثوابت , IB , QB , MB , VB , AC , SMB

شرح كل ما يمكن كتابته على مفاتيح المقارنة:

م	الأنواع	التوضيح
١	ثوابت	قيم تتم كتابتها أثناء البرمجة وهى غير قابلة للتغير أثناء عمل البرنامج.
٢	IB	هى عبارة عن مجموعة من ثمان مفاتيح.
٣	IW	هى عبارة عن مجموعة من ستة عشر مفتاح.
٤	ID	هى عبارة عن مجموعة من أثنين وثلاثين مفتاح.
٥	QB	هى عبارة عن مجموعة من ثمان مخرجات.

٦	QW	هى عبارة عن مجموعة من ستة عشر خرج.
٧	QD	هى عبارة عن مجموعة من أثنين وثلاثين خرج.
٨	MB	هى عبارة عن مجموعة من ثمان ريلهات.
٩	MW	هى عبارة عن مجموعة من ستة عشر ريليه.
١٠	MD	هى عبارة عن مجموعة من أثنين وثلاثين ريليه.
١١	VB	هى عبارة عن متغيرات بحجم byte.
١٢	VW	هى عبارة عن متغيرات بحجم word.
١٣	VD	هى عبارة عن متغيرات بحجم Dword.
١٤	SMB	هى عبارة عن مجموعة من ثمان ريليهات خاصة.
١٥	SMW	هى عبارة عن مجموعة من ستة عشر ريليه خاص.
١٦	SMD	هى عبارة عن مجموعة من أثنين وثلاثين ريليه خاص.
١٧	AC	هو عبارة عن محتوى للقيام بالعمليات الحسابية.

أمثلة عملية:

١- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث المحرك الأول يعمل بعد ثلاث ثوانى من ضغط المفتاح بينما يعمل المحرك الثانى بعد خمس ثوانى من الضغط على نفس المفتاح باستخدام مؤقت زمنى واحد.

٢- قم بتنفيذ دائرة تحكم منطقية للمبتان فلاشر بحيث يعملان بالتبادل فتعمل اللمبة الأولى لمدة ثانية بنما تبقى الأخرى مطفأة لمدة ثانية وهكذا.

٣- قم بتنفيذ دائرة تحكم منطقية لمحرك يعمل لليمين لزمن ثم يقف زمن أخر و يعمل يساراً لزمن ثم يقف لزمن أخر وهكذا.

المثال الأول:

✓ لمحركين بحيث المحرك الأول يعمل بعد ثلاث ثواني من ضغط المفتاح بينما يعمل المحرك الثاني بعد خمس ثواني من لضغط على نفس المفتاح باستخدام مؤقت زمني واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	==I	T50
٢	==I	T50
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.4/K1M
٢	كونتاكتور	Q0.7/K1M

الشرح:

:Network1

يالضغط على I0.1 يبدأ المؤقت الزمني T50 بالعمل.

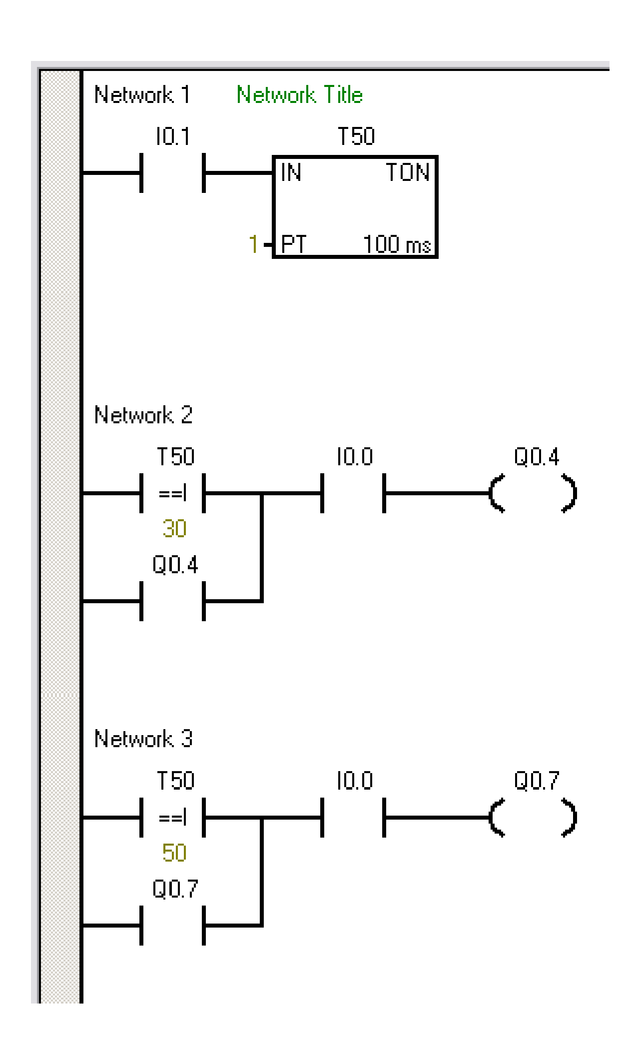
:Network2

عندما يصل المؤقت الزمني إلى ثلاث ثواني يغلق مفتاح المقارنة فيعمل المحرك Q0.4

:Network3

عندما يصل المؤقت الزمني إلى خمس ثواني يغلق مفتاح المقارنة فيعمل المحرك Q0.7

البرنامج:


Network 1
Network Title
I0.1
T50
IN
TON
1
PT
100 ms
Network 2
T50
==I
30
I0.0
Q0.4
Q0.4
Network 3
T50
==I
50
I0.0
Q0.7
Q0.7

المثال الثانى:

✓ لمبتان فلاشر بحيث يعملان بالتبادل فتعمل اللمبة الأولى لمدة ثانية بنما تبقى الأخرى مطفأة لمدة ثانية وهكذا.

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.c.	I0.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	<=I	T33
٢	<=I	T33
٣	>=I	T33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q0.4/K1M
٢	لمبة	Q0.7/K1M

الشرح:

:Network1

يالضغط على I0.1 يبدأ المؤقت الزمنى T33 بالعمل.

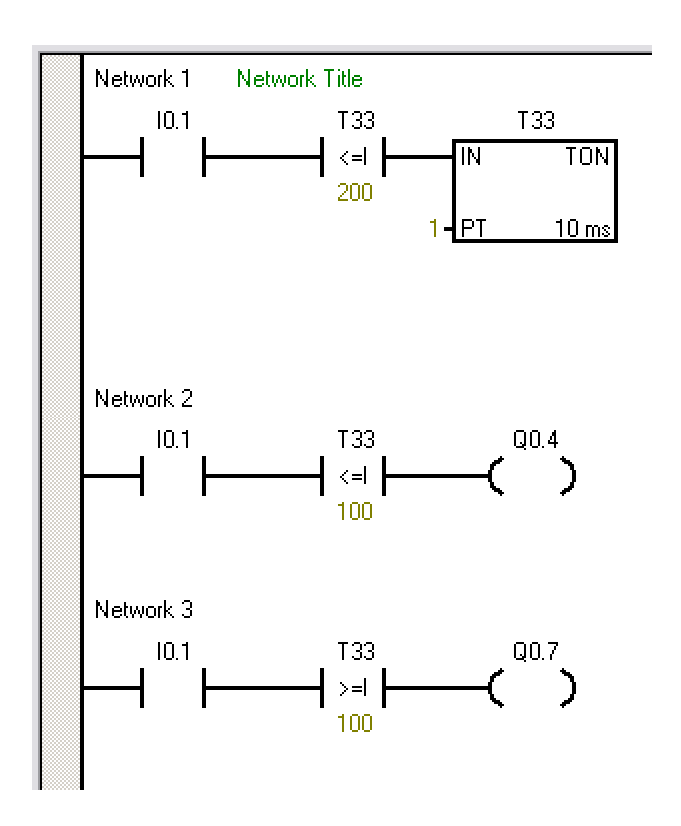
:Network2

بالضغط على I0.1 تبدأ Q0.4 بالعمل حتى أن يتعدى المؤقت الزمنى الثانية.

:Network3

تبدأ Q0.7 بالعمل عندما يتعدى المؤقت الزمنى ثانية وعندا يصل المؤقت الزمنى إلى ثانيتان يتوقف المؤقت الزمنى ليبدأ من جديد.

البرنامج:


Network 1
Network Title
I0.1
T33
<=I
200
T33
IN
TON
1
PT
10 ms
Network 2
I0.1
T33
<=I
100
Q0.4
Network 3
I0.1
T33
>=I
100
Q0.7

المثال الثالث:

- ✓ لمحرك يعمل لليمين لزمن ثم يقف زمن أخر و يعمل يساراً لزمن ثم يقف لزمن أخر وهكذا.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I1.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	>=I	T32
٢	<=I	T32
٣	<=I	T32
٤	<=I	T32
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.2/K1M
٢	كونتاكتور	Q3.1/K1M

الشرح:

Network1:

بالضغط على I1.1 يبدأ المؤقت الزمني T32 بالعمل.

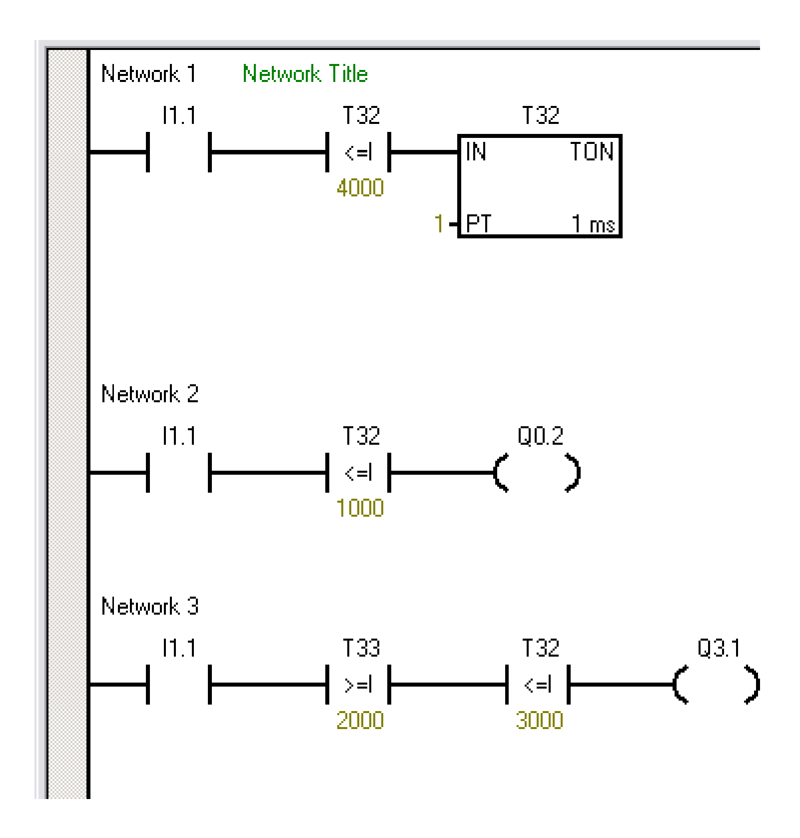
Network2:

بالضغط على I1.1 تبدأ Q0.2 بالعمل حتى أن يتعدى المؤقت الزمني الثانية.

:Network3

تبدأ Q3.1 بالعمل عندما يتعدى المؤقت الزمني الثانيتان وعندا يصل المؤقت الزمني إلى ثلاث ثوانى يتوقف المؤقت الزمني لمدة ثانية أخرى ليبدأ من جديد.

البرنامج:

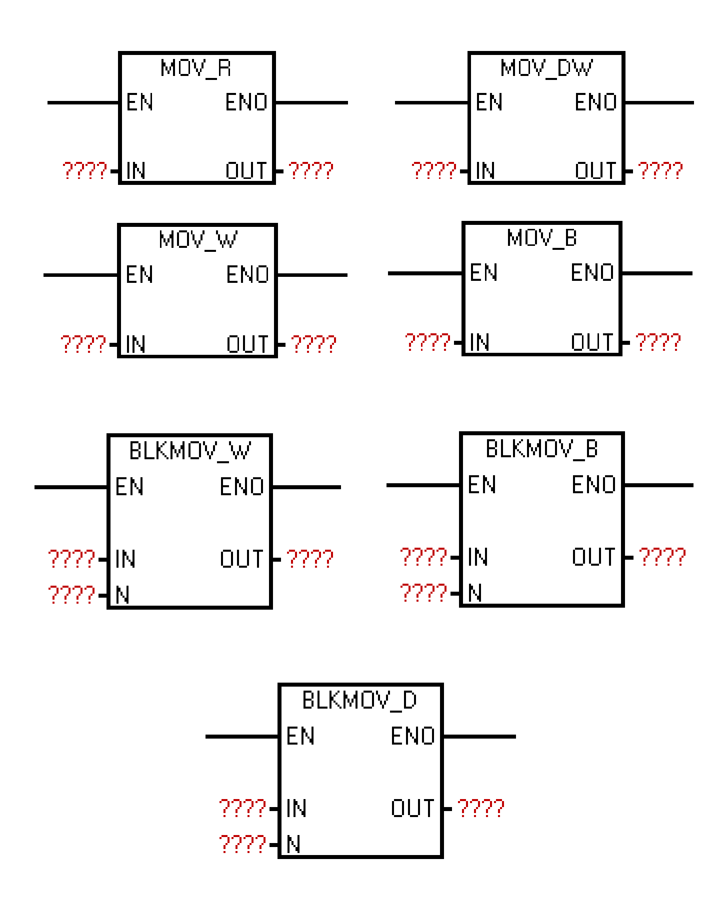


الباب التاسع

عمليات النقل

- أنواع عمليات النقـــــــــــل داخل الـــ PLC.
- عمليات نقـــل للأرقام الصحيحة و للأرقام العشرية.
- عمليات نقـــــــــــــــــل لحجم الـــ byte.
- عمليات نقـــــــــــل مجموعات بحجم الـــ byte.
- عمليات نقـــــــــــــــــل لحجم الـــ word.
- عمليات نقـــــــــــل مجموعات بحجم الـــ word.
- عمليات نقل لحجم الـــ Dword للأرقام الصحيحة.
- عمليات نقـــل لحجم الـــ Dword للأرقام العشرية.
- عمليات نقـــــــــــل مجموعات بحجم الـــ Dword.
- تماريـــــــن تـــطـــبـــيـــقـــيـــة على عمليات النقل.

عمليات نقل القيم.......MOVE :



- تستخدم عمليات نقل القيم MOVE لنقل أى قيمة من داخل أى ذاكرة إلى أى ذاكرة أخرى مع مراعاة أن يكون حجم الذاكرة التي سوف يتم نقل القيمة لها هو نفس حجم للذاكرة التي تم نقل القيمة منها.

- تستخدم عمليات نقل المجموعات BLKMOVE لنقل أى عدد من Byte أو Word أو D.word مع مراعاة أن يكون حجم الذاكرة التى سوف يتم نقل القيمة لها هو نفس حجم الذاكرة التى تم نقل القيمة منها.
- يوجد سهم على يمين الـــ MOVE أو BLKMOVE يعمل كمفتاح يغلق عندما يتم تنفيذ العملية المراده.

أنواع عمليات النقل:

م	الأسم	الشرح	الشكل
١	عمليات النقل Byte	عمليات نقل الـــ Byte تستخدم فى نقل أى ثوابت (أرقام صحيحة) أو متغيرات من مشتقات الـــ Byte إلى أى متغيرات أخرى من مشتقات الـــ Byte.	MOV_B EN ENO ????- IN OUT -????
٢	عمليات النقل Word	عمليات نقل الـــ Word تستخدم فى نقل أى ثوابت (أرقام صحيحة فقط) أو متغيرات من	

		مشـــتقات الـــــ Word إلى أى متغيرات أخرى من مشتقات الـــــ Word.	MOV_W EN ENO ????-IN OUT-????
٣	عمليات النقل Dword	عمليات نقــل الـــــ D.Word تستخدم فى نقل أى ثوابت (أرقـــام صحيحة فقط) أو مـــتغيرات مـــن مشـــتقات الـــــ D.Word إلى أى متغيرات أخرى مـــن مشـــتقات الـــ D.Word.	MOV_DW EN ENO ????-IN OUT-????
٤	عمليات النقل Real	عمليات نقــل الـــــ D.Word تستخدم فى نقل أى ثوابت (أرقـــام عشرية فقـــط) أو مـــتغيرات مـــن مشـــتقات الـــــ D.Word إلى أى متغيرات أخرى مـــن مشـــتقات الـــ D.Word.	MOV_R EN ENO ????-IN OUT-????

أنواع عمليات نقل المجموعات:

م	الأسم	الشرح	الشكل
١	عمليات نقل مجموعة Byte	عمليات نقل لمجموعة الـــــ Byte تســـتخدم فى نقل أى عدد من الـــــ Byte بشـــرط أن تكون متتالية فى الترتيـــب. يمكـــن أن تكـــون مجموعة الـــ Byte تحتـــوى علـــى ثوابـــت (أرقام صحيحة فقط) أو أى مـــتغيرات مـــن مشـــتقات الـــــ Byte إلى أى مـــتغيرات أخرى من مشتقات الـــ Byte.	BLKMOV_B EN ENO ???? IN OUT ???? ???? N
٢	عمليات نقل مجموعة Word	عمليـــات نقـــل لمجموعـــة الـــــ Word تستخدم فى نقل أى عدد مـــن الـــــ Word بشرط أن تكون متتالية فى الترتيـــب. يمكـــن أن تكون مجموعـــة الـــــ Word تحتـــوى	BLKMOV_W EN ENO ???? IN OUT ???? ???? N

		على ثوابت (أرقـــام صـــحيحة فقــط) أو أى متغيرات مــن مشــتقات الــــ Word إلى أى متغيرات أخــرى مــن مشــتقات الـــ Word.	
٣	عمليات نقل لمجموعة Dword	عمليـــات نقـــل لمجموعـــة الـــ DWord تســـتخدم فى نقـــل أى عـــدد مـــن الـــ DWord بشـــرط أن تكـــون متتاليـــة فى الترتيب. يمكـــن أن تكـــون مجموعـــة الـــ DWord تحتـــوى علـــى ثوابـــت (أرقـــام صحيحة أو أرقام عشـــرية) أو أى مـــتغيرات مـــن مشـــتقات الـــ DWord إلى أى مـــتغيرات أخـــرى مـــن مشـــتقات الـــ DWord.	BLKMOV_D EN ENO ??? IN OUT ???? ??? N

كل ما يمكن كتابته فى دخل أو خرج عمليات النقل:

م	النوع	المكان	الرموز المستخدمة
١	MOV_B EN ENO ????-IN OUT-????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , IB , QB , AC , SMB , VB , MB
		مسميات الخرج OUT	QB AC , SMB , VB , MB ,
٢	MOV_W EN ENO ????-IN OUT-????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , IW , QW , AC , SMW , VW , MW
		مسميات الخرج OUT	, SMW , VW , MW , QW AC
٣	MOV_DW EN ENO ????-IN OUT-????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , ID , QD , AC , SMD , VD , MD
		مسميات الخرج OUT	QD AC , SMD , VD , MD ,
٤	MOV_R EN ENO ????-IN OUT-????	مسميات الدخل IN	ثوابت(أرقام عشرية فقط) , ID , QD , AC , SMD , VD , MD
		مسميات الخرج OUT	QD AC , SMD , VD , MD ,

كل ما يمكن كتابته فى دخل أو خرج عمليات نقل المجموعات:

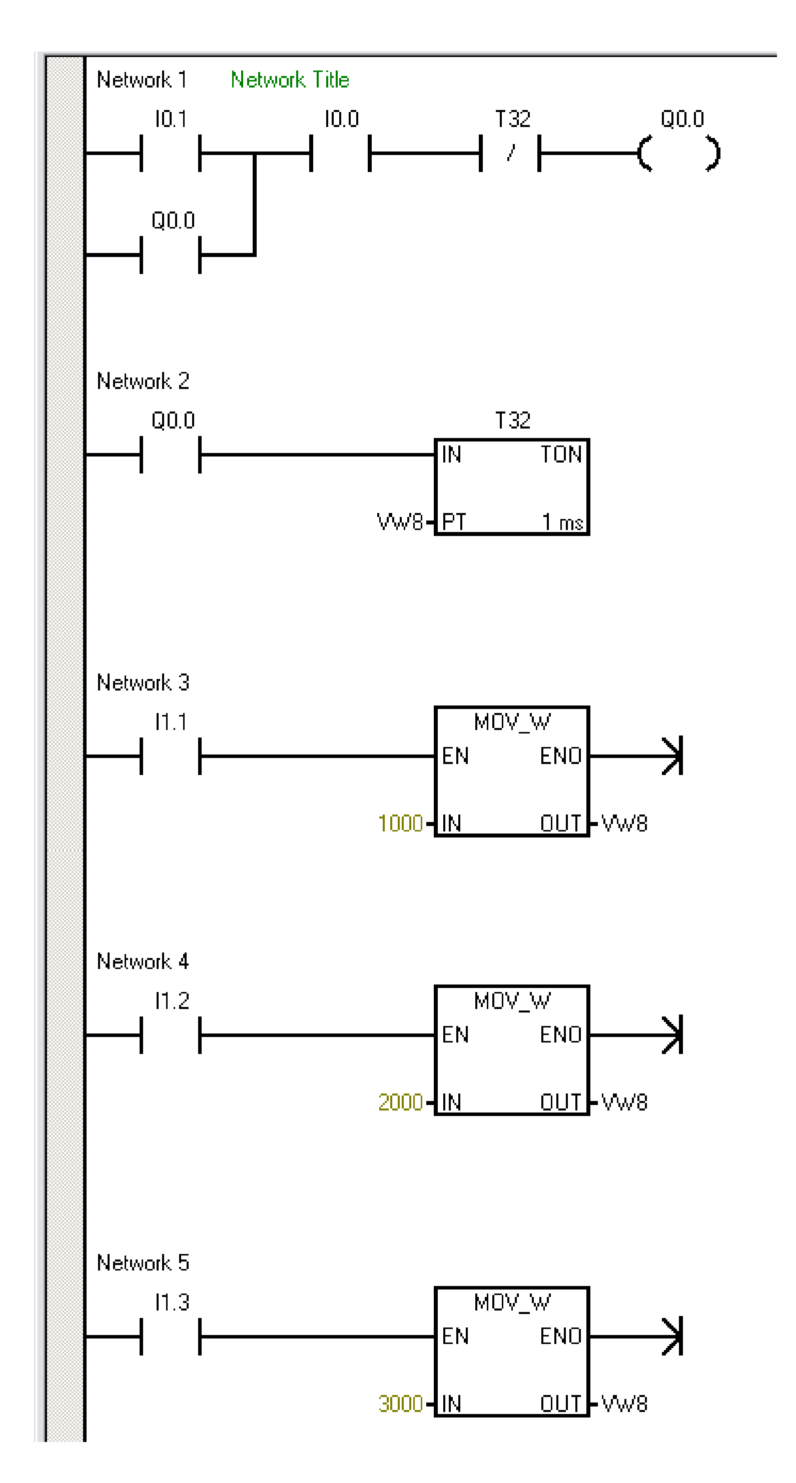
م	النوع	المكان	الرموز المستخدمة
١	BLKMOV_B EN ENO ???? IN OUT ???? ???? N	مسميات الدخل IN	IB , QB , MB , VB , SMB , AC
		مسميات الخرج OUT	QB , MB , VB , SMB , AC
		N	من ١ إلى ٢٥٥
٢	BLKMOV_W EN ENO ???? IN OUT ???? ???? N	مسميات الدخل IN	IW , QW , MW , VW , SMW , AC
		مسميات الخرج OUT	QW , MW , VW , SMW , AC
		N	من ١ إلى ٢٥٥
٣	BLKMOV_D EN ENO ???? IN OUT ???? ???? N	مسميات الدخل IN	ID , QD , MD , VD , SMD , AC
		مسميات الخرج OUT	QD , MD , VD , SMD , AC
		N	من ١ إلى ٢٥٥

أمثلة (تمارين عملية):

١– قم بتنفيذ دائرة تحكم منطقية لمحرك يعمل لمدة زمن ثم يقف, بشرط أن يكون هذا الزمن متغير (١ثانية – ٢ثانية – ٣ثانية).

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I1.1/S3
٤	n.o.	I1.2/S4
٥	n.o.	I1.3/S5
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد عمليات النقل	**نوع عمليات النقل**	**أسم عمليات النقل**
١	Word	MOV_W
٢	Word	MOV_W
٣	Word	MOV_W
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

التمرين:


Network 1
Network Title
I0.1
I0.0
T32
Q0.0
Q0.0
Network 2
Q0.0
T32
IN
TON
VW8
PT
1 ms
Network 3
I1.1
MOV_W
EN
ENO
1000
IN
OUT
VW8
Network 4
I1.2
MOV_W
EN
ENO
2000
IN
OUT
VW8
Network 5
I1.3
MOV_W
EN
ENO
3000
IN
OUT
VW8

الشرح:

:Network1

بالضغط على I0.1 يعمل الخرج Q0.0 إلى أن يتم فصلة بواسطة مفتاح الإيقاف أو بواسطة النقطة المغلقة الخاصة بالمؤقت الزمني T32.

:Network2

عندما يعمل الخرج يبدأ المؤقت الزمني T32 بالعمل.

:Network3

بالضغط على I1.1 تصبح قيمة الـــ VW8 هي ١٠٠٠ فيعمل المؤقت الزمني لثانية واحدة.

:Network4

بالضغط على I1.2 تصبح قيمة الـــ VW8 هي ٢٠٠٠ فيعمل المؤقت الزمني لثانيتان.

:Network5

بالضغط على I1.2 تصبح قيمة الـــ VW8 هي ٣٠٠٠ فيعمل المؤقت الزمني لثلاث ثواني.

الفهرس

الفهرس

الفهرس

الفهرس

الكتب التي صدرت عن معهد السالزيان الإيطالى "دون بوسكو"

الكتاب	المؤلف
🕮 **محركات, مولدات و محولات التيار المتردد**	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الأول	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الثاني	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الأول	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الثاني	وجيه جرجس
🕮 **الدوائر العملية للضغوط الهوائية و الكهروهوائية**	وجيه جرجس
🕮 **غسالة الأطباق**	وجيه جرجس
🕮 **زانوسى الموديلات القديمة 14-16-18بروجرام**	وجيه جرجس
🕮 **الدوائر الكهربائية للتركيبات المترلية**	نبيل رزق
🕮 **صيانة وإصلاح الأجهزة المترلية**	نبيل رزق
🕮 **أفكار التكيف و التبريد للدوائر الميكانيكية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد للدوائر الكهربائية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد الخدمة والأعطال**	إميل فتح الله
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الأول	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الثاني	تحت التحضير
🕮 **موديلات الغسالة كريازى**	تحت التحضير